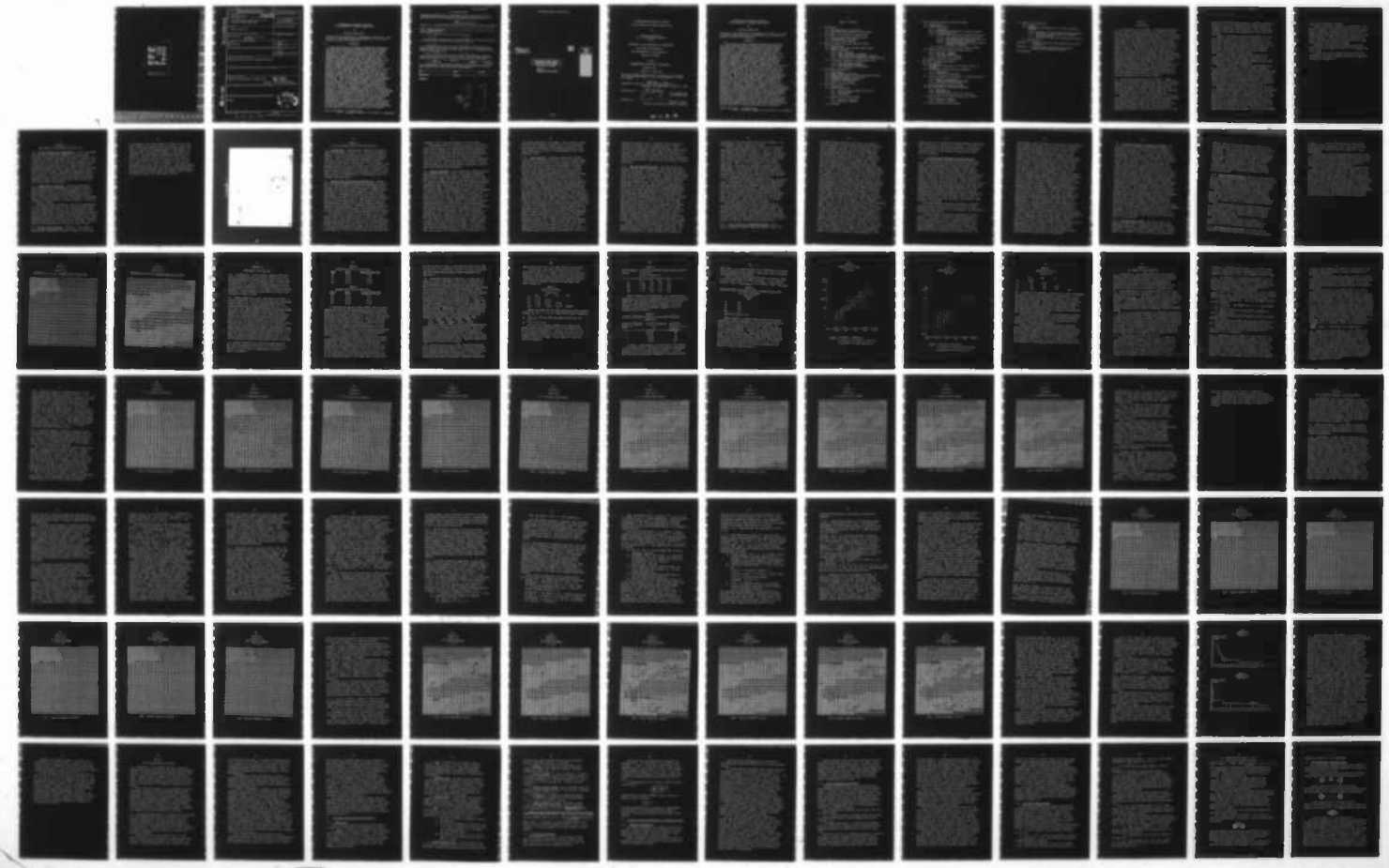
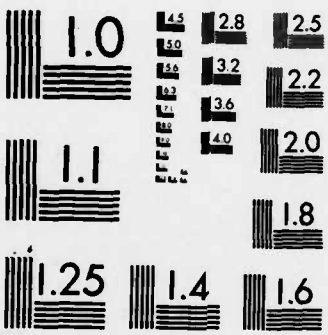
AD-A139 124 A CONSTRAINED-CLUSTERING APPROACH TO THE ANALYSIS OF REMOTE SENSING DATA(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH C J RYAN 1983 1/2

UNCLASSIFIED AFIT/CI/NR-83-80T F/G 12/1 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

UNCLASS

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

AD A139124

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFIT/CI/NR 83-80T	2. GOVT ACCESSION NO: A139124
3. RECIPIENT'S CATALOG NUMBER:	
4. TITLE (and Subtitle): A Constrained-Clustering Approach to the Analysis of Remote Sensing Data	5. TYPE OF REPORT & PERIOD COVERED: THESIS/DISSERTATION
	6. PERFORMING ORG. REPORT NUMBER:
7. AUTHOR(s): Charles Joseph Ryan	8. CONTRACT OR GRANT NUMBER(s):
9. PERFORMING ORGANIZATION NAME AND ADDRESS: AFIT STUDENT AT: Massachusetts Institute of Technology	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS:
11. CONTROLLING OFFICE NAME AND ADDRESS: AFIT/NR WPAFB OH 45433	12. REPORT DATE: 1983
	13. NUMBER OF PAGES: 133
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office):	15. SECURITY CLASS. (of this report): UNCLASS
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

APPROVED FOR PUBLIC RELEASE: IAW AFR 190-17

10 Feb 1984

LYNN E. WOLAVER
Dean for Research and
Professional Development

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

ATTACHED

DTIC ELECTE MAR 2 0 1984 S D A

DTIC FILE COPY

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASS

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

84 03 19 050

A CONSTRAINED-CLUSTERING APPROACH TO THE ANALYSIS OF REMOTE SENSING DATA

by

Charles Joseph Ryan

Submitted to the Department of Mathematics on January 13, 1983 in partial fulfillment of the requirements for the Degree of Master of Science in Mathematics

ABSTRACT

One old and two new clustering methods were applied to the constrained-clustering problem of separating different agricultural fields based on multispectral remote sensing satellite data. (Constrained-clustering involves double clustering in multispectral measurement similarity and geographical location.) The results of applying the three methods are provided along with a discussion of their relative strengths and weaknesses and a detailed description of their algorithms. The classical K-Means Clustering method was applied to the data to provide a baseline upon which to compare the two new methods, since the K-Means method's properties are well known. Weaknesses of this method include a requirement to specify the number of clusters as input to the algorithm, the algorithm's search for and formation of spherical clusters, the resulting locally (not globally) optimum solution, and failure of the algorithm to consider generality of its application and its well known properties. The Kth Nearest Neighbor method was also applied to the data. This new procedure clustered the multispectral data as accurately as the K-Means method. Its weaknesses include failure to consider geographical information, uncertainty on the final choice of K and the disposition of "left-over" elements. The method's strengths include its ability to estimate the number of clusters in the underlying population. The Graph Decomposition method is a computationally efficient graph-partitioning procedure for solving a constrained-clustering problem modeled as a graph network. Its resulting clusters were as good as or better than the other two methods. A weakness of the method is its inability to determine the exact number of clusters present until a density contour threshold value is somewhat subjectively chosen by the analyst. Its strengths include the output of the tree of high-density clusters and geographical location information considerations. It does not require a priori the number of clusters present and can be applied to a much more general class of problems.

Thesis Supervisor: M. Anthony Wong
Title: Assistant Professor, Sloan School of Management

AFIT/CI/NR 83-80T

AFIT RESEARCH ASSESSMENT

The purpose of this questionnaire is to ascertain the value and/or contribution of research accomplished by students or faculty of the Air Force Institute of Technology (ATC). It would be greatly appreciated if you would complete the following questionnaire and return it to:

AFIT/NR
Wright-Patterson AFB OH 45433

RESEARCH TITLE: A Constrained-Clustering Approach to the Analysis of Remote Sensing Data

AUTHOR: Charles Joseph Ryan

RESEARCH ASSESSMENT QUESTIONS:

1. Did this research contribute to a current Air Force project?

( ) a. YES ( ) b. NO

2. Do you believe this research topic is significant enough that it would have been researched (or contracted) by your organization or another agency if AFIT had not?

( ) a. YES ( ) b. NO

3. The benefits of AFIT research can often be expressed by the equivalent value that your agency achieved/received by virtue of AFIT performing the research. Can you estimate what this research would have cost if it had been accomplished under contract or if it had been done in-house in terms of manpower and/or dollars?

( ) a. MAN-YEARS _____ ( ) b. $_____

4. Often it is not possible to attach equivalent dollar values to research, although the results of the research may, in fact, be important. Whether or not you were able to establish an equivalent value for this research (3. above), what is your estimate of its significance?

( ) a. HIGHLY SIGNIFICANT ( ) b. SIGNIFICANT ( ) c. SLIGHTLY SIGNIFICANT ( ) d. OF NO SIGNIFICANCE

5. AFIT welcomes any further comments you may have on the above questions, or any additional details concerning the current application, future potential, or other value of this research. Please use the bottom part of this questionnaire for your statement(s).

NAME GRADE POSITION

ORGANIZATION LOCATION

STATEMENT(s):



FOLD DOWN ON OUTSIDE - SEAL WITH TAPE

AFIT/NR
WRIGHT-PATTERSON AFB OH 45433

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE, $300

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 73236 WASHINGTON D.C.

POSTAGE WILL BE PAID BY ADDRESSEE

AFIT/ DAA
Wright-Patterson AFB OH 45433

FOLD IN

A CONSTRAINED-CLUSTERING APPROACH TO THE ANALYSIS OF REMOTE SENSING DATA

by

Charles Joseph Ryan
B.S., University of Maryland
(1975)

SUBMITTED TO THE DEPARTMENT OF MATHEMATICS

IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS OF THE DEGREE OF

MASTER OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

January 1983

c Charles J. Ryan 1983

The author hereby grants to M.I.T. permission to reproduce and to distribute copies of this thesis document in whole or in part.

Signature of Author.....Charles J. Ryan.....
Department of Mathematics, January 13, 1983

Certified by.....M. A. Wong.....
M. Anthony Wong
Thesis Supervisor

Accepted by.....
Nesmith C. Ankeny
Chairman, Departmental Graduate Committee

84 03 19 050

A CONSTRAINED-CLUSTERING APPROACH TO THE ANALYSIS OF REMOTE SENSING DATA

by

Charles Joseph Ryan

Submitted to the Department of Mathematics on January 13, 1983 in partial fulfillment of the requirements for the Degree of Master of Science in Mathematics

ABSTRACT

One old and two new clustering methods were applied to the constrained-clustering problem of separating different agricultural fields based on multispectral remote sensing satellite data. (Constrained-clustering involves double clustering in multispectral measurement similarity and geographical location.) The results of applying the three methods are provided along with a discussion of their relative strengths and weaknesses and a detailed description of their algorithms. The classical K-Means Clustering method was applied to the data to provide a baseline upon which to compare the two new methods, since the K-Means method's properties are well known. Weaknesses of this method include a requirement to specify the number of clusters as input to the algorithm, the algorithm's search for and formation of spherical clusters, the resulting locally (not globally) optimum solution, and failure of the algorithm to consider generality of its application and its well known properties. The Kth Nearest Neighbor method was also applied to the data. This new procedure clustered the multispectral data as accurately as the K-Means method. Its weaknesses include failure to consider geographical information, uncertainty on the final choice of K and the disposition of 'left-over' elements. The method's strengths include its ability to estimate the number of clusters in the underlying population. The Graph Decomposition method is a computationally efficient graph-partitioning procedure for solving a constrained-clustering problem modeled as a graph network. Its resulting clusters were as good as or better than the other two methods. A weakness of the method is its inability to determine the exact number of clusters present until a density contour threshold value is somewhat subjectively chosen by the analyst. Its strengths include the output of the tree of high-density clusters and geographical location information considerations. It does not require a priori the number of clusters present and can be applied to a much more general class of problems.

Thesis Supervisor: M. Anthony Wong
Title: Assistant Professor, Sloan School of Management

Table of Contents

1. Introduction

- 1.1 The Objectives of this Study
- 1.2 The Outline and Contents of this Report

2. The Landsat II Multispectral Imagery Data

- 2.1 The Data Source and Storage Medium
- 2.2 A Description of the Data
- 2.3 The Data Image Analyzed

3. Previous Analysis and the Proposed Methods

- 3.0 Introduction
- 3.1 General Goals and Practical Applications
- 3.2 Previous Analyses
  - 3.2.1 Their Approach
  - 3.2.2 Their Preliminary Analysis
  - 3.2.3 Their Findings in Subsequent Analyses
  - 3.2.4 Limitations and Problems with their Approach
- 3.3 A Different Approach to Analyzing the Data
  - 3.3.1 The Underlying Concept
  - 3.3.2 Three Proposed Procedures
  - 3.3.3 The Objectives of this Study

4. Preliminary Data Analysis

- 4.0 Introduction
- 4.1 Image Summary Statistics
- 4.2 Summary Statistics for the Two Selected Regions
- 4.3 Dimensional Analysis of the Data
- 4.4 Principal Component Analysis
- 4.5 Summary of the Findings

5. The K-Means Clustering Method

- 5.0 Introduction
- 5.1 Method Description
  - 5.1.1 The Criterion (Minimize Trace (w))
  - 5.1.2 The Algorithm
  - 5.1.3 Properties
- 5.2 Applying the K-Means Method
  - 5.2.1 Data Set 1 Results
  - 5.2.2 Data Set 2 Results
- 5.3 Problems

6. The Kth Nearest Neighbor Clustering Method

6.0 Introduction
6.1 Method Description

6.1.1 Background
6.1.2 A Theoretical Approach to Forming Clusters
6.1.3 Literature Survey
6.1.4 The Kth Nearest Neighbor Clustering Algorithm

6.1.4.1 The Density Estimation Algorithm
6.1.4.2 The Hierarchical Clustering Algorithm
6.1.4.3 The Computational Algorithm
6.1.4.4 Strong Set-Consistency of Kth Nearest Neighbor Clustering

6.1.5 Diagnostic Plots for the Number of Modes

6.2 Applying the Kth Nearest Neighbor Method

6.21. Data Set 1 Results
6.2.2 Data Set 2 Results
6.2.3 Diagnostic Plot Analysis

6.3 Method Evaluation and Problems

7. The Graph Decomposition Method

7.0 Introduction
7.1 The Method's Application to a General Class of Problems
7.2 Preliminary Definitions and Discussions

7.2.1 Definitions
7.2.2 Preliminary Discussion

7.3 Existing Clustering Techniques (A Literature Survey)

7.3.1 Summary of Categories
7.3.2 Graph Decomposition Techniques
7.3.3 The Single Linkage Method

7.4 The High-Density Clustering Model
7.5 The High-Density Clustering Algorithm

7.5.1 The Goals
7.5.2 The Algorithm
7.5.3 Comments on the Model and Algorithm

7.6 Application of the Graph Decomposition Method

7.6.1 Data Set 1 Results
7.6.2 Data Set 2 Results

7.7 Method Evaluation and Problems

8. Summary and Conclusions

8.1 Summary
8.2 Strengths and Weaknesses of the Three Clustering Methods

8.2.1 Evaluation of the K-Means Clustering Method
8.2.2 Evaluation of the Kth Nearest Neighbor Clustering Method
8.2.3 Evaluation of the Graph Decomposition Method

8.3 Conclusions

Appendix A -- Examples using the Graph Decomposition Method of Chapter 7

Appendix B -- Mortran Program Implementing the Graph Decomposition Method of Chapter 7

Bibliography

Chapter 1

Introduction

1.1. The Objectives of this Study: "Constrained-clustering" problems, that is, problems requiring clustering of objects in both general attribute measurements and in spatial relationships, can often be modeled by a graph network. In this study, the problem of crop classification based on multispectral measurements of large agricultural areas from the Landsat II satellite will be approached from a "constrained-clustering" problem viewpoint. The overall scope and complexity of the task will be analyzed and limited based on the perpective gained from previous work on the problem. Once the obstacles have been identified and the scope sufficiently limited, three clustering methods will be evaluated and their properties discussed in relation to their ability to address this "constrained-clustering" problem. One of these methods, the K-Means Clustering Method is well known and studied. The other two, the K^{th} Nearest Neighbor Clustering Method and the Graph Decomposition Method are new and offer powerful, distinct advantages over the K-means method.

1.2. The Outline and Contents of this Report: Chapter 2 contains the multispectral imagery data source and a detailed description of the data, including when, where, and how the data was collected. Two subsets of the data image are selected for detailed analysis. These subsets are specified, and the reasons for their selection are also given.

Chapter 3 provides a detailed description of a previous analysis done on similar data and outlines the three proposed clustering methods for further analysis and comparison. This chapter includes reasons for collecting the data and practical applications which result from its analysis. Following a discussion of a previous study done in this area

along with its results and problems, a different approach to aalyzing the data is proposed along with the underlying concepts, assumptions, and objectives. Three alternative procedures for applying this approach are then identified briefly.

Chapter 4 provides a preliminary analysis on the two data subsets chosen for this study. This analysis was performed to describe the overall characteristics of the data and to verify several conclusions of previous work in this area. The preliminary analysis includes summary statistics, dimensional analysis, principal component analysis, and a summary of the results.

Chapter 5 contains a detailed description of the classical K-Means Clustering Method, the results of applying this method to the selected subsets of data, and a discussion of the method's problems.

Similarly, Chapter 6 contains a detailed description of a new clustering method. The K^{th} Nearest Neighbor Clustering Method is discussed including pertinent background and theoretical preliminaries, a literature survey of competing techniques with their deficiencies, and a complete specification of the implementation algorithm. The method is then applied to the two subsets of data, the results are provided, and the associated problems identified.

Chapter 7 contains a detailed discussion of the other new clustering method proposed for analyzing the "constrained-clustering" problem based on a graph network model. This Graph Decomposition Method is developed in detail including definitions, a literature survey of related methods and historics, the development of the high-density clustering model, and the presentation of a computationally efficient algorithm for applying the method. The results of its application on the same two subsets of data is also included

along with the method's problems.

Chapter 8 provides a summary of the findings in the previous chapters. It discusses the strengths and weaknesses of the three proposed methods for solving the "constrained-clustering" problem so that they can be more easily compared. Lastly, this chapter highlights the study's conclusions including the applicability of the three methods to more general problems.

Appendix A includes several examples using the Graph Decomposition Method to show its effectiveness and ease of use.

Finally, the actual computer program listing for implementing the Graph Decomposition Method's algorithm is included as Appendix B.

Chapter 2
The Landsat II Multispectral Imagery Data

2.1. The Data Source and Storage Medium: In the last decade there has been considerable interest in analyzing data collected by airborne and spaceborne sensors to identify and inventory the earth's resources. One such collection of data, the one used in this study, was provided on magnetic tape by the System and Services Division of Lockheed Electronics Company Incorporated, Houston, Texas 77058. Their nine-track, 800 bytes per inch (bpi), IBM computer compatible magnetic tape is used to store the Landsat II multispectral imagery data from two 5 × 6 nautical mile Large Area Crop Inventory Experiment (LACIE) sites, namely Finney County, Kansas and Grand Forks County, North Dakota.

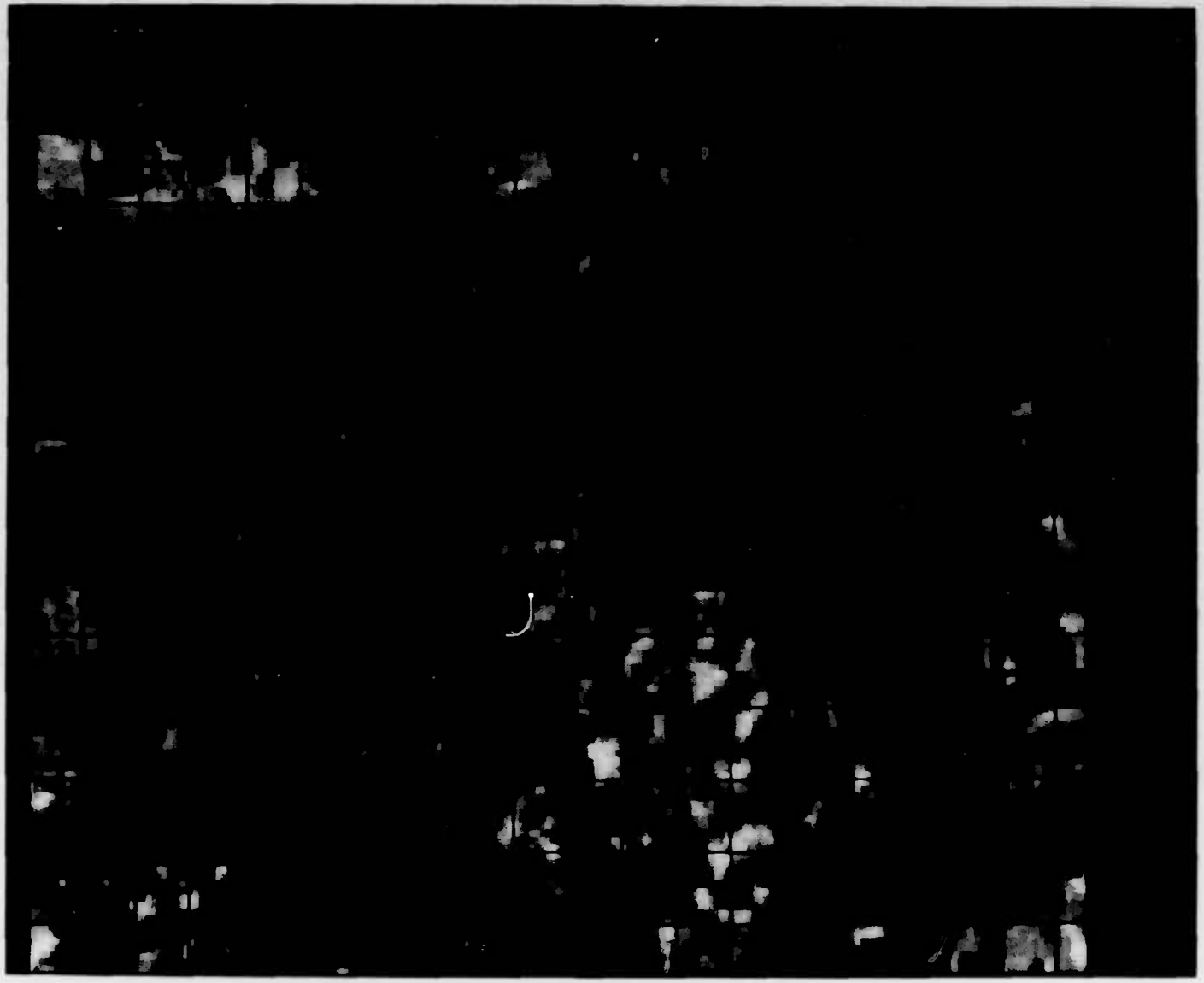
2.2. A Description of the Data: The data tape contains eleven images of the Finney County site taken at various times between October 22, 1975 and September 28, 1976, and twelve images of the Grank Forks County acquired between May 6, 1976 and September 10, 1976. Each image is 196 pixels (ground resolution elements or picture elements) in width and 117 pixels in length.

According to Misra and Wheeler (1978, 1980) measurements of light intensity were made in four visable and near infra-red wave length bands, resulting in four dimensional, multi-spectral imagery data for each pixel and providing a repetitive coverage of the earth's surface at the two LACIE sites. These measurements appear on the data tape as calibrated, digitized radiance measured counts for each of the four bands. The four bands will be labeled B1, B2, B3 and B4 in what follows.

2.3. The Data Image Analyzed: The June 12, 1976 image of Finney County, Kansas was arbitrarily chosen for this study.

A photograph of this image appears as Figure 1 on the next page. Because of computer storage limitations (there were only 32, 768 words of available memory in the HP3000), two twenty pixels by twenty pixels subsets of this image were analyzed. The first subset, subsequently referred to as Data Set 1 consists of pixel rows 31 to 50 and columns 25 to 44. The second, Data Set 2, consists of rows 71 to 90 and columns 31 to 50. These two subsets of the image are bordered by heavy dark lines in Figure 1. They were selected for this study because each contains several geographically situated groups of pixels easily identified by color (different shades of gray in the photograph, Figure 1).

FIGURE 1



Chapter 3
Previous Analysis and the Proposed Methods

3.0. Introduction: Satellite data consisting of spectral images of the earth's surface, e.g., the data described in Chapter 2, has been collected for many different practical reasons. There are many possible approaches to analyzing this type of data and the choice is somewhat dependent on the particular goal or application. In this chapter, some possible uses of this type of imagery data are first given and then a previous analysis of this data is reported. The goals, finding, limitations, and conclusions of this previous study serve as an introduction to, and motivation for, the analysis undertaken here.

3.1. General Goals and Practical Applications: Imagery data have been collected by airborne and spaceborne sensors for more than ten years to monitor our environment and to identify and survey the earth's resources. Weather systems observation is one well known example of monitoring the environment. Multispectral satellite data have also been used to monitor crops, land use, water and mineral resources. Implicit in these applications is the assumption that the electromagnetic radiance from a ground scene of different objects is distinguishable. Therefore, one basic purpose of such data collection and analysis is the identification of distinct objects within the sample multispectral data images.

As indicated in Chapter 2, the data set in this study can be used for crop identification; in fact, Misra and Wheeler (1978, 1980) maintain that one of the most important and promising applications of such data is in providing crop product inventories over large areas. The goal is therefore to classify the multispectral imagery data to determine the total extend of a crop in a given ground scene and eventually

to predict total crop yields based on all such scenes.

The approach would be to develop accurate crop classification rules based on the premise that different fields of a given crop of interest in a particular area are physically similar and therefore have similar spectral characteristics, i.e., each individual crop has its own unique spectral signature distinguishing it from all other crops, vegetation, and ground covers. The effectiveness of developed classification schemes can then be obtained by comparing the results with actual ground truth information.

3.2. Previous Analyses: In the recent past, several authors have analyzed this type of data. The recently completed LACIE project is one such example. According to Misra and Wheeler (1978, 1980), the aim of this project was to analyze the Landsat multispectral imagery data to provide crop product inventories over large areas. The project was broken into three basic tasks: stratification of an area of interest into domains which were homogeneous relative to the proportion of crop area and/or yield; estimation of the extent and average yield of the crop in each domain on the basis of the satellite data; and, finally, aggregation to predict the total crop yields on a global scale. To this end, crop area estimates obtained by classification of the multispectral scanner data from Landsat were to be combined with information from meteorological satellites to estimate crop product inventories. Misra and Wheeler (1978, 1980) addressed the second basic task. Their goal was to classify the multispectral imagery to determine the amount of a specific crop in a given scene by estimating the areas in Landsat images occupied by the crop or crops of interest. Their approach, although quite general, will be discussed in relation to their work on the identification of wheat. The results of their work serve as an introduction to, and motivation for

the problem addressed in this paper. Other relevant issues involving the remaining two basic tasks, such as sampling, data labeling procedures, area proportion estimation, and yield models, may be found in the Proceedings of the LACIE Symposium (1978).

3.2.1. Their Approach: Analogous discriminant analysis, classification and identification of crops in Landsat images requires a characterization of their "spectral signatures" and a decision rule to match the measurements of unknown objects to the known signatures of the crops of interest. The end result would be the development of mathematical models of the spectral phenomenon and classification schemes to provide crop classification algorithms based on the promise that different fields of a crop in a given area are physically similar and, therefore, have similar spectral characteristics. These similar spectral characteristics of a crop, based on some metric and the available ground truth data, are called its spectral signature and were to be Misra and Wheeler's basis for crop classification. Their spectral similarity model of crop signatures was based on a statistical model for the data. This model, accepted by default, hypothesized that observations from a given field of a crop constitute a sample from a probability distribution characteristic of its crop type. The probability law is usually assumed Gaussian, characterized by a mean vector and a dispersion matrix. The spectral signature consists of estimates of these parameters based on known (training) data for the crop(s) of interest. The data for their study consisted of mean vectors and dispersion matrices for a number of known wheat and nonwheat fields from several sites. Each site had data from observations at several different times of the year; each such observation is called an acquisition of the site. Under the statistical model, the data from the homogeneous fields were

assumed multivariate normal, independent from one pixel to the next with common, known covariance matrix. These assumptions are generally violated by the data, leaving their test results asymptotic at best. Their hope was that crop signatures characterized by fields within a few miles of each other would also be valid over a larger area, permitting global characterization, classification, and product inventory estimation.

3.2.2. Their Preliminary Analysis: The preliminary data analysis done by Misra and Wheeler (1978, 1980) resulted in several key findings. First, the data from the agricultural sites was essentially two-dimensional. The data from different sites and from the same sites taken at different times of the year (an "acquisition") lie on parallel planes in the four-dimensional feature space. This finding not only permits deviations of the data from the plane to be interpreted as noise, but also allows the data to be displayed graphically by taking two linear combinations of the four-channel data and transforming the mean vectors in a rotated coordinate frame of the two dimensions; both aid the analysis. To reach their conclusion of two-dimensional data, they performed a chi-squared test where the test statistic for the pooled fields was partitioned into a sum of components for testing dimensionality, parallelness, and equality of the planes. They performed a canonical analysis of the mean vectors and covariance matrices from measurements of known, similar fields. Geometric arguments based on the mean vectors and principal component analysis of the dispersion matrix of the pooled data from all acquisitions lead to the same conclusion.

Second, a finding which supports the first, they found the within-field variability of the data to be small. The standard deviations in the four channels ranged from .9 to 3.5 in all acquisitions. These acquisitions included all

four phases of the wheat growth cycle (crop calendar): crop establishment, green, heading, and mature.

Finally, and most importantly, they found the spectral measurements from different fields of a crop in a Landsat image showed considerable variability. However, the commonly used crop classification algorithms are based on the premise that different fields of a crop are physically similar and have similar spectral characteristics for each Landsat acquisition. The ℓ_1, ℓ_2, or weighted ℓ_2 metrics are often used to determine the "spectral closeness" of such measurements. The found great field-to-field variability in the sense of these metrics. This large variability arose naturally from differences in planting dates, soil characteristics, treatments, and weather, to name a few.

The third finding lead to a breakdown of their statistical model. For example, the average values of the observations from wheat fields could not reasonably have been drawn from the same probability distribution. Hypothesis tests for equality of mean vectors and dispersion matrices across wheat fields continually failed at each stage of the growth cycle even when the correlation structure of the data was taken into account. These tests failed due to the large field-to-field variability in each crop phase.

Mirsa and Wheeler's preliminary analysis found no adequate characterization of a crop's signature based on its spectral response from randomly selected fields. Their spectral similarity model was unsupported by the date from the acquisitions studied. They found no strong separability between the spectral measurements from the wheat and nonwheat fields nor from wheat and the various ground covers.

3.2.3. Their Findings in Subsequent Analyses: Given their preliminary findings, Misra and Wheeler (1978, 1980)

performed further analyses on the data in an attempt to discover features of the spectral response that would provide a signature of wheat. For each acquisition, they plotted against each other the two linear combinations of the transformed mean vectors of a field. The path chronologically connecting the plots of all acquisitions for a given field constitutes a "temporal trajectory" in the place spanned by the two-dimensional data. These plots are simple devices for displaying the multi-date spectral measurements and their pattern of temporal change for each crop. An examination of these multitemporal plots for a number of sites revealed that in each case the pattern of temporal change in spectral response characterizes the crop, provides a valid signature, and is a basis for crop classification in Landsat images (the considerable field-to-field variability in spectral measurements notwithstanding). The measurements from the wheat fields showed significant differences at each stage, but in each case the pattern of change showed certain characteristic features. The shape of each sampled trajectory depended on sampling times and often appeared different for two sites with acquisitions at different times in the crop calendar. Also, atmospheric conditions at the time of an acquisition affects the trajectory shape. However, for a given site acquisition, these conditions did not mask the distinguishing features in patterns of temporal change in spectral response of the different ground covers, since any distortion would be common to all covers. Even though different fields of a crop frequently appeared spectrally quite dissimilar throughout the growth cycle, the crop's temporal pattern of change, when tied to the crop's calendar, was similar for each field. This stamp of the crop calendar was so evident that Misra and Wheeler (1978, 1980) suggested a new approach to crop recognition based on the pattern of change in the measurements. Their

approach calls for a measure of distance between trajectories and a comparison of their angular features. The problem of crop classification would then effectively be transformed into one similar to the problem of recognition of hand-written characters.

3.2.4. Limitations and Problems with Their Approach:

Misra and Wheeler's approach has several limitations and other remaining problems which require further study. First, Landsat measurements of spectral response and, therefore, temporal trajectories depend on many factors including crop calendars, planting dates, data sampling times, atmospheric conditions during sampling, soil characteristics, treatments, local weather conditions, precipitation history, sun angle, and viewing angle. These sources of difficulty are some of the causes for the large field-to-field variability within and between acquisitions. They make crop identification difficult and weaken separability of crop measurements from different ground covers in a scene by comparing either spectral responses or temporal trajectories. To make matters worse, wheat and barley are so similar in physical appearance and crop calendars, that they are often confused on the ground.

The second problem with Misra and Wheeler's approach, which is closely related to the first, is that it cannot distinguish wheat and oats and is inadequate in distinguishing between wheat and barley.

A third problem, in large part responsible for the second, is the number and timing of data acquisitions are important, sometimes critical, to adequate delineation of the temporal patterns of the crops of interest. Small, subtle differences in crop calendars may be captured or missed depending on available acquisitions. For example, without one critical acquisition at a specific time in the

crop calendar, wheat and barley are totally indistinguishable. This problem would prevent the success of their approach if that particular acquisition was not acquired due to cloud cover at the time Landsat passed over. In fact, their results were not satisfactory partly because they were forced to work with a minimal set of labeled, training data. Fewer data acquisitions compounded their problem because of the "nearness" of the temporal patterns of several other ground covers. Their decision rules, then, need to take into account the field-to-field variability of the crops of interest and must not be dependent on certain time critical acquisitions. Improved decision rules will depend upon an increased understanding of the reflectance characteristics of crops' vegetative canopies at various stages of growth and levels of vigor depending on such things as treatment and weather conditions. They'll also depend on an understanding of the effects of measurements by atmospheric conditions (haze, for example), irradiance of the scene, viewing angle, and brightness of the scene (sun angle). A much more detailed study will be required to satisfactorily understand the phenomenological interpretations of these measurements. A better understanding would lead to the development of mathematical models of these phenomenon and classification schemes based on a system of ground truth (training) data. Adequate training data would consist of a systematic account of ground observations from agricultural areas to record the status of fields, temperature, precipitation history, atmospheric conditions, and others. However, the extraction of training data in Landsat images requires human intervention, including spatial structure and correlation considerations, which is costly, slow, and error prone. These human tasks are essential to an operational crop inventory program; along with the problems mentioned above, they highlight the limitations of the approach.

3.3. A Different Approach to Analyzing the Data: The ultimate goal is to provide more precise, reasonably economical algorithms for extracting crop information from remotely sensed data. In light of the problems and complexities of the approach described above, the scope of this study is less ambitious, less general, more modest, and much more economical in terms of time, cost, and the amount of data required. Unlike Misra and Wheeler's (1978, 1980) approach attempting to establish a crop's signature based on its spectral response from many data acquisitions of several agricultural sites over the crop's entire growth calendar, this study will analyze two subsets of the data from a single acquisition of one particular site by using, evaluating, and comparing three methods of identifying like objects in a Landsat scene. These methods include one new and two classical clustering algorithms (described below). By restricting this study to two small areas of only one site acquisition, a number of necessary assumptions have been simplified and many others eliminated. The many factors that affect spectral responses such as sampling times, crop calendar, atmospheric conditions, precipitation history, sun angle, and viewing angle are either not applicable or constant. Therefore, a crop should have nearly the same physical appearance and spectral response throughout a field, and because spatial information is considered, the variability between fields caused by such factors as planting dates, soil characteristics, and treatments is less important.

3.3.1. The Underlying Concept: The main concept to be employed in this study is clustering in both the spectral measurement space and in geographical contiguity, or double clustering. Along with spectral response, spatial grouping needs to be included in clustering pixels into homogeneous

fields. Spatial grouping can be done in several ways. The simplest way is to cluster spectrally and then cluster spatially. Only contiguous pixels would then remain in a cluster. In the ideal case, each cluster would actually be a field. A somewhat more involved grouping procedure would combine and split on spatial as well as spectral "nearness". The result might be a proliferation of clusters, an obvious disadvantage of these procedures. In addressing this possibility, the number of spectral clusters and/or certain threshold parameters will have to be investigated.

3.3.2. Three Proposed Procedures: The aim of each procedure is to identify contiguous batches of pixels with similar multispectral measurements. This is the "constrained-clustering problem." Clustering is performed to find groups of pixels with similar multispectral measurements, but these groups or clusters are "constrained" by spatial considerations.

The first procedure partitions the four-dimensional measurements into K clusters using the classical "K-means" method. Then, contiguous batches of pixels will be extracted from the K clusters to form the "geographically-constrained" clusters.

The second procedure uses the recently developed (Wong and Lane, 1983) K^{th} nearest neighbor clustering algorithm to determine the number of clusters (modes) in the multispectral measurement space. Spatial clusters are formed based on the number of modes.

The final procedure involves a new approach. It includes a computationally-efficient graph-decomposition method for solving the constrained-clustering problem.

3.3.3. The Objectives of this Study: The first task is to perform a preliminary analysis of the data image described

in Section 2.3 in order to verify some of the general findings of Misra and Wheeler (1978, 1980) such as its two-dimensional structure and the variability of the measurements.

The remaining tasks include describing in detail each of the constrained-clustering methods, applying them to the two subsets of data defined in Section 2.3, presenting the results, and evaluating and comparing the methods by identifying their strengths and weaknesses.

There is no available ground truth data against which to compare the results of each method. Instead, the red and non-red (mostly green) areas in Data Sets 1 and 2 are shown in Figures 2 and 3 respectively. The boundaries of these areas were determined by inspection of a multispectra color photograph of the site like the black and white version in Figure 1, but not included in this report. It was not clear into which area some of the boundary pixels belonged. In fact, many of the area boundaries were fuzzy and decisions were very subjective. This should be kept in mind when evaluating the results of the three clustering methods.

Figure 2

Data Set 1

(Hatched areas are red, unshaded areas are non-red)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120
121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140
141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160
161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180
181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200
201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220
221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240
241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260
261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280
281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300
301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320
321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340
341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360
361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380
381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400

Figure 3

Data Set 2

(Hatched areas are red, unshaded areas are non-red)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120
121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140
141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160
161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180
181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200
201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220
221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240
241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260
261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280
281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300
301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320
321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340
341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360
361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380
381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400

Chapter 4

Preliminary Data Analysis

4.0. Introduction: In this chapter, summary statistics will be presented for both the entire image and its two subsets selected for further analysis (both are described in paragraph 2.3). Next, the dimensionality of the agricultural data is examined; a principal component analysis is performed on Data Set 2 in an attempt to verify previous findings. A brief summary of the results and their implications will also be given.

4.1. Image Summary Statistics: The means of the data in the four bands (B1, B2, B3 and B4) for the whole image were 39.98, 50.80, 59.03, and 25.14 respectively, and the standard deviations of these same bands were 5.02, 7.66, 7.29, and 3.48. These computed statistics correspond almost perfectly with those reported by the data source (see Chapter 2). Misra and Wheeler (1978, 1980) reported that within-filed, between pixels standard deviations ranged from .9 to 3.5 for the four bands in all acquisitions. These two sets of figures are not directly comparable since the first set of standard deviations is for the entire image pooled across pixels and fields. Nevertheless, these figures indicate the between fields standard deviations are more than twice those between pixels within a field. This larger variability between fields indicates there is hope of distinguishing different fields based solely on the data.

4.2. Summary Statistics for the Two Selected Regions: The means and standard deviations of the four bands for Data Set 1 and Data Set 2, respectively, are given in Table 1 and Table 2.

TABLE 1

Summary Statistics for Data Set 1

Band	Mean	Standard Deviation
B1	40.43	2.38
B2	50.27	3.00
B3	55.63	4.94
B4	23.07	2.79

TABLE 2

Summary Statistics for Data Set 2

Band	Mean	Standard Deviation
B1	37.74	2.15
B2	48.02	3.26
B3	58.62	8.03
B4	25.69	4.49

A comparison of the two tables reveals that the means between the two sets are approximately within one standard deviation of each other, while the standard deviations are generally much higher for Data Set 2. In fact, the standard deviations for the last two bands (B3 and B4) in Data Set 2 are nearly double those of Data Set 1. That this should be the case can be seen by examining Figure 1. The top set (Data Set 1) appears much more homogeneous in color than the bottom set. In fact, Data Set 2 was partly chosen because of its several, distinct, well defined patterns, and Data Set 1 was chosen because it has only two.

A similar comparison between the tables and the means and standard deviations of the entire image shows no major differences. The means for the selected subsets are roughly within one standard deviation of the image means. Image means are nearer to those of Data Set 1 for B1 and B2 and closer to those of Data Set 2 for B3 and B4. It is also interesting to note that the standard deviations for all bands are higher for the image than for Data Set 1,

but for B3 and B4, those of the image are lower than those of Data Set 2. As expected, Data Set 1 is more homogeneous than the image, and the image roughly comparable with Data Set 2.

4.3. Dimensional Analysis of the Data: As mentioned in Chapter 3, Misra and Wheeler (1978, 1980) concluded that the multispectral data are essentially two-dimensional. Data from all the different sites and different acquisitions lay on parallel planes in the four dimensional feature space. They based their conclusion on a canonical analysis of the mean vectors and covariance matrices of known, similar fields and on two key observations. First, the within-field variability was generally small. Second, for the three classes of fields studied (wheat, non-wheat, and pooled) the field means lay on or close to two-dimensional, parallel planes. Essentially, they have reduced the four-channel data into two without losing too much information. And their transformation of the mean vectors into two dimensions involved two linear combinations (LCs):

$$LC1 = .406B1 + .600B2 + .645B3 + .234B4$$

$$LC2 = -.386B1 - .530B2 + .535B3 + .532B4$$

Kauth and Thomas (1976) have provided phenomenological interpretations of these linear combinations. They suggested that LC1 and LC2 can be interpreted roughly as measures of brightness and greenness. LC2 measures the contrast in average measurements in the visible (B1 and B2) and near-infrared (B3 and B4) bands. The more vigorous the vegetation, the larger the contrast.

4.4. Principal Component Analysis: The two-dimensional data given in Misra and Wheeler (1978, 1980) are obtained by a principal component analysis of the dispersion matrix of the pooled data from any acquisition. It has been shown that the first two principal components reconstruct

the original data with a very high degree of efficiency.

To this end, a principal component analysis was performed on Data Set 2. Table 3 contains the covariance matrix for this data set. The variances of the bands are consistent with the standard deviations reported earlier. (The Principal Component analysis was performed using BMDP Program 4M).

TABLE 3

Covariance Matrix

Data Set 2

	B1	B2	B3	B4
B1	4.61			
B2	4.41	10.66		
B3	4.80	12.51	64.43	
B4	1.70	5.45	34.49	20.19

Table 4 contains the Principal Components factor loading vectors $\underset{\sim}{a}_i$ for $i = 1,2,3,4$ where the principal components $y_i = \sum_{j=1}^{4} a_{ij} \cdot B_j$; the B_j's are the bands B1,B2,B3,B4; and $\underset{\sim}{a}_i = (a_{11}, a_{12}, a_{13}, a_{14})$.

The first two principal components loading factor vectors show little similarity with linear combinations LC1 and LC2 of paragraph 4.3. However, Data Set 2 is not a whole image on acquisition and wouldn't be expected to conform to the whole image. It was chosen for its distinctive patterns.

TABLE 4

Factor Loading Vector	Principal Component Factor Loading Vectors Data Set 2 Factor Loading for			
	B1	B2	B3	B4
$\underset{\sim}{a}_1$	.07	.18	.86	.47
$\underset{\sim}{a}_2$	.49	.83	-.09	-.28
$\underset{\sim}{a}_3$	.86	-.51	.07	-.05
$\underset{\sim}{a}_4$	.15	.09	-.49	.85

Table 5 contains the eigenvalues (λ_i) and cumulative proportions of the total variance explained by each principal component. The variance explained by each principal component is the eigenvalue for that component. The total variance is the sum of the diagonal elements of the covariance matrix of Table 3.

TABLE 5

Principal Components Proportion of Variance Explained
Data Set 2

Principal Component Number (i)	Variance Explained (λ_i)	Cumulative Proportion of Total Variance
1	86.148	.86
2	10.402	.97
3	2.236	.99
4	1.099	1.00

As Table 5 clearly shows, the data does in fact appear to be two-dimensional. The first two principal components account for 97% of the total system variance. In fact, when the first two principal components are plotted against

each other for all observations (not included here) two easily distinguishable clusters appear. These facts tend to confirm Misra and Wheeler's conclusion.

Table 6 contains the squared multiple correlations (SMC) of each variable with all other variables. In the familiar notation $SMC_1 = R^2_{B1\text{-}B2,B3,B4}$ or in general $SMC_i = R^2_{Bi \cdot Rest}$.

TABLE 6

Squared Multiple Correlations (SMC) of Each Variable With All Other Variables

Data Set 2

	SMC
B1	.41
B2	.51
B3	.93
B4	.93

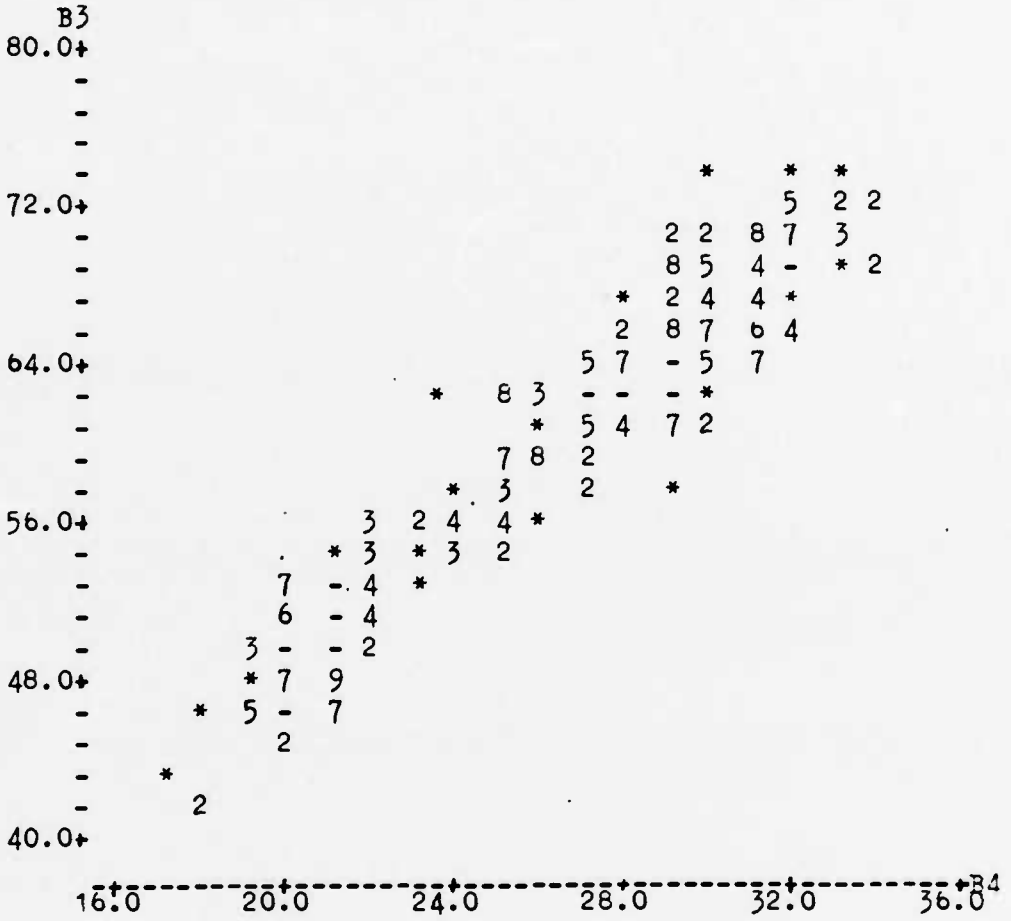
Condition number = 78.4

Table 6 appears to indicate that B3 and B4 are highly correlated (linearly dependent) and the correlation matrix in Table 7 tends to confirm this with correl (B3,B4) = .96. Figure 4 is a scatterplot of B3 and B4. Again, they appear to be linearly dependent. Figure 5 is the same scatterplot of B3 and B4 for Data Set 1. Here too, the results are the same. As a final confirmation, a linear regression was performed on B3 versus B4 using BMDP Program IR. This resulted in a Multiple R of .96 and a Multiple R^2 of .92. In fact, B1 and B2 were also found to be linearly related, but not nearly as strong as were B3 and B4.

FIGURE 4

SCATTERPLOT OF B3 VS B4

DATA SET 2



Legend: * represents 1 observation

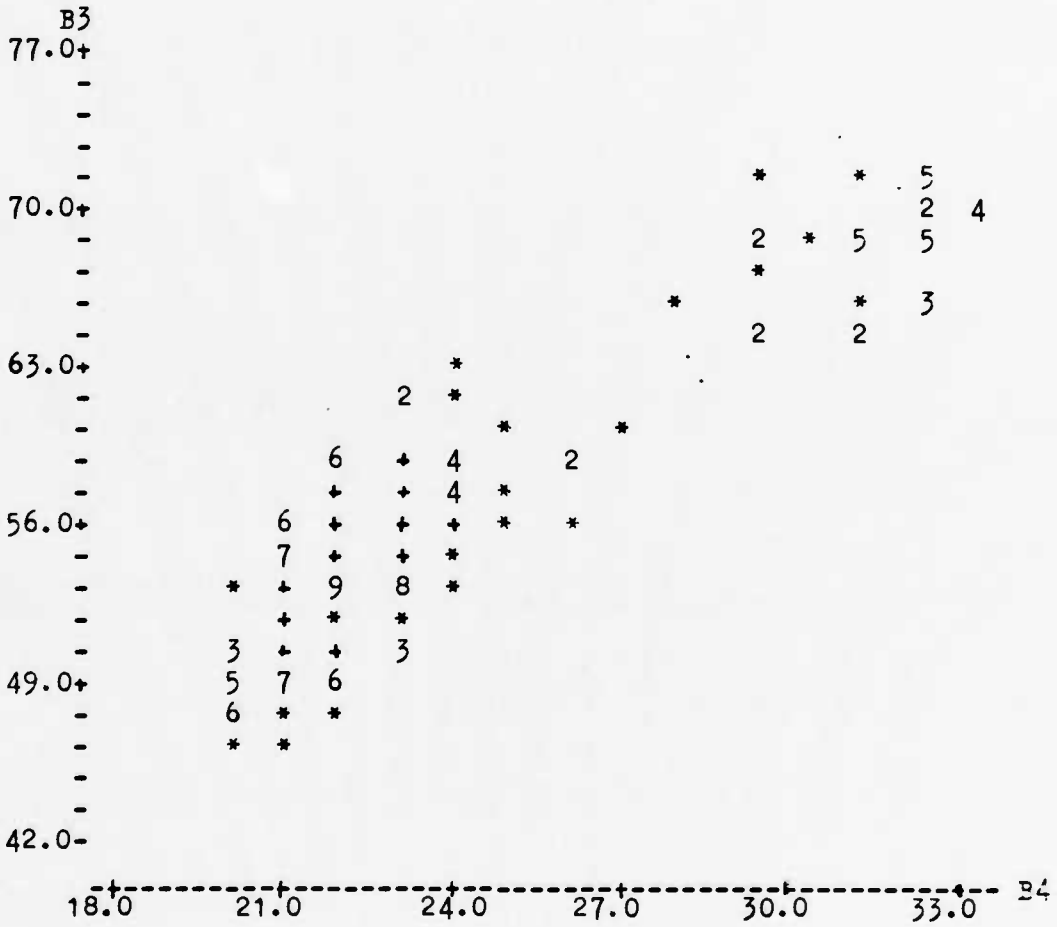
1-9 represent this number of observations

+ represents 10 or more observations

FIGURE 5

SCATTERPLOT OF B3 VS B4

DATA SET 1



Legend: * represents 1 observation

1-9 represent this number of observations

+ represents 10 or more observations

TABLE 7
Correlation Matrix
Data Set 2

	B1	B2	B3	B4
B1	1.00			
B2	.63	1.00		
B3	.28	.48	1.00	
B4	.18	.37	.96	1.00

4.5 Summary of the Findings: Several results from preliminary analyses of the data were presented in this chapter. First, the between fields variability was found to be much greater than that within fields. This was true between all classes of fields (wheat, nonwheat, and pooled). This result supports the hope that different fields can be distinguished from each other based on the data alone or possibly in conjunction with geographical considerations.

Second, our four-channel data can be effectively reduced to only two dimensions, which also have meaningful physical interpretations. This reduction will allow the data to be depicted in two-dimensional displays.

Finally, the principal components analysis verified Misra and Wheeler's conclusions and found that bands B3 and B4 were linearly dependent, and that B1 and B2 were also somewhat linearly related. The importance of this finding lies in its implied suggestion to try two different bandwidths not linearly related to B1 or B3, or to remove one or two of the bandwidths to save future collection and processing costs.

Chapter 5
The K-Means Clustering Method

5.0 Introduction: This chapter describes the classical K-Means clustering method, provides the results of applying this method to the two data sets for various values of K and discusses the method's strengths and weaknesses in analyzing imagery data. For a detailed discussion of this method, see Hartigan (1975) and Hartigan and Wong (1979).

5.1 Method Description: The K-Means method is one of the most popular partitioning techniques. It partitions n vector observations $\underline{X}_1, \underline{X}_2, \ldots, \underline{X}_n$ into K compact, convex, spherical clusters. The K-Means algorithm attempts to minimize trace (w) where w is the pooled within clusters sum of squares matrix. From an euclidean distance point of view, it attempts to minimize distances within groups and maximize distances between groups.

It is not practicable to examine all possible K-partitions to find the global optimum. Instead, some approximate heuristic is often used to find a local optimum. For example, a partition is defined to have neighboring partitions which can be obtained from it by moving an observation from one group of the partition to another. A local optimum is defined as a partition which has a smaller trace (w) than its neighbors and is reached by moving observations to any neighbor with a smaller trace (w).

5.1.1 The Criterion (Minimize Trace (w)):

For observations $\underline{X}_1, \underline{X}_2, \ldots \underline{X}_n$, let $X_{i,j}$ be the ith observation of the jth variable. To construct a partition of the n observations into K clusters, define a partition function P, for the ith observation, P(i) = 1,2,3... or K for $1 \leq i \leq n$. Let $\underline{C}_1, \underline{C}_2, \ldots \underline{C}_K$ be a set of cluster centers for a partition. Then the partition P and its cluster sum of squares (trace (2) by trace (2) = $\sum_{i,j} (X_{i,j} - C_{p(i),j})^2$.

The goal is to minimize this quantity or at least to make it small. Other criteria are sometimes used (e.g. Wilk's lambda and Hotelling's trace) see Friedman and Rubin (1967).

5.1.2 The Algorithm: As previously mentioned, it is not practical to evaluate all possible partitions in order to obtain the global minimum of trace (w). Instead, there are many approximating techniques. The algorithm below is called the "K-Means" (or ISODATA) technique:

1. Begin with an initial partition P or initial cluster centers $\underline{C}_1, \ldots \underline{C}_K$. There are many ways to do this, including random assignment and distances from the overall mean.

2. Given P, compute $\underline{C}_1, \ldots \underline{C}_K$ to be the means of points in the K clusters.

3. Given $\underline{C}_1, \ldots \underline{C}_K$, define $P(i) = j$ if $\underline{X}_i$ closest to $\underline{C}_j$, $1 \leq j \leq K$.

4. Return to step 2 if the partition P changes in step 3; otherwise stop.

An efficient computational algorithm is given in Hartigan and Wong (1979).

This algorithm guarantees that each pass through steps 2 and 3 reduces trace (w), so that cycling does not occur.

However, there is no guarantee that the global optimum will be reached, and very little is known about how to identify it if it is reached.

5.1.3 Properties: Almost nothing is known about how to determine when the global optimum is reached. In fact, the locally optimal solutions obtained from different initial starting configurations are often different. In addition, the clusters are not invariant with respect to linear transformations, so that weighting of the variances can change the results. Using trace (w) as the criterion gives equal weights to each of the variables involved, but other schemes

of scaling the variables are possible. For equally weighted variables, those with greater variability dominate the formation of clusters.

Hartigan (1975) shows that K-Means clusters are convex polygons for both local and global optimums, and he provides a more detailed discussion of their shapes. In fact the K-Means algorithm implicitly assumes clusters are spherical structures.

Another property of the K-Means technique (a major drawback of most partitioning techniques) is that K is generally not known before hand. Hartigan (1975) recommends the number of clusters K should not be decided in advance, but the algorithm should be run with several different values of K. Then, analysis of variance, on each of the variables, for each clustering will help decide which number of clusters is best. He also suggests computing covariances within each cluster, the overall covariance matrix within clusters, and the overall covariance matrix between clusters as additional aids. One heuristic method of choosing K proposed by Hartigan (1975) involves calculating the within cluster sum of squares for K clusters (WSS_K) and then plotting $K \frac{\overline{WSS_K}}{N-K}$ versus K and looking for an elbow point.

5.2 Applying the K-Means Method: In this section the K-Means Clustering method described above is applied to the two data sets to solve the constrained clustering problem. The goal is to identify contiguous batches of pixels with similar multispectral measurements. To this end the 400 4-dimensional observations (pixels) were partitioned into K clusters using the K-Means method provided in the BMDPKM program. Once these K clusters were formed, contiguous batches of pixels belonging to the same cluster were used to represent the "geographically-constrained" clusters (agricultural fields).

5.2.1 Data Set 1 Results: Figures 6 through 10 show these constrained clusters for Data Set 1 for K = 2,3 ... 6, respectively using the K-Means method. Comparison of these figures with Figure 2 shows that for each K, the first cluster (cluster A) identifies the red (hatched) area perfectly. As K increases from 2 to 6, the pixels bordering the red area change clusters. This is an indication of the gradual change from red to non-red (generally green) in the original color picture or from light to dark in Figure 1. Because of a lack of other additional identifiable structure in Figure 1, no other comments can be made about increasing K from 2 to 6. Each cluster represents a different shade of gray (e.g. for K = 6, cluster C represents the darkest shades of gray).

5.2.2 Data Set 2 Results: Figures 11 through 15 show the geographically-constrained clusters for Data Set 2 for K = 2,3,4,5, and 6 respectively using the K-Means method. Since there is a much more detailed structure in Figure 3 than in Figure 2, a comparison of Figure 3 with Figures 11 through 15 will also be more detailed.

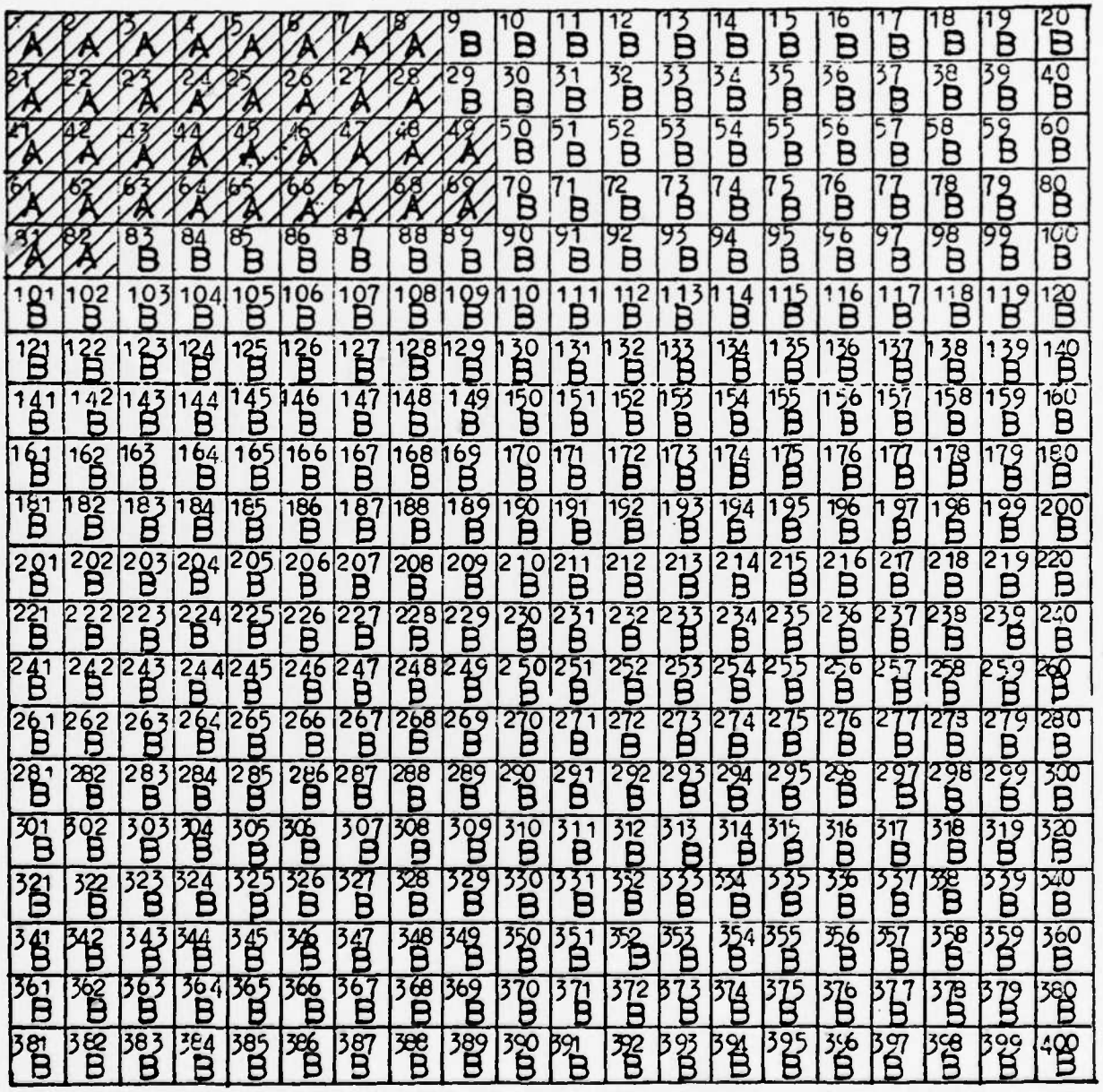
For K = 2 in Figure 11, there are several pixels which appear to deviate from the pattern of Figure 3. These pixels are numbered 5, 176, 223-224, 275, 290, 290-291, 376-380, and 392-400. Pixels 5, 376-380, and 392-400 are in Cluster A but aren't red (hatched). This is due to the fact that there are only two clusters and these pixels are light, as seen in Figure 1. The remaining "mismatched" pixels are on the border between red and non-red. Such classifications were done subjectively by eye, and there was often doubt as to which category a pixel belonged.

For K = 3 in Figure 12, pixel 176 remains a "mismatched" border pixel. Here, "mismatched" means the subjective classifications in Figure 3 don't match the K-Means

Figure 6

Data Set 1

K = 2 clusters from K-Means



Note: letters represent clusters

Figure 7

Data Set 1

K = 3 clusters from K-Means

1 A	2 A	3 A	4 A	5 A	6 A	7 A	8 A	9 C	10 C	11 C	12 C	13 C	14 C	15 C	16 C	17 C	18 C	19 C	20 C
21 A	22 A	23 A	24 A	25 A	26 A	27 A	28 A	29 B	30 C	31 C	32 C	33 C	34 C	35 C	36 C	37 C	38 C	39 C	40 C
41 A	42 A	43 A	44 A	45 A	46 A	47 A	48 A	49 A	50 C	51 C	52 C	53 C	54 B	55 B	56 C	57 C	58 C	59 C	60 B
61 A	62 A	63 A	64 A	65 A	66 A	67 A	68 A	69 A	70 B	71 C	72 B	73 B	74 B	75 B	76 B	77 B	78 C	79 B	80 C
81 A	82 A	83 B	84 C	85 C	86 C	87 C	88 C	89 C	90 C	91 C	92 C	93 B	94 B	95 B	96 B	97 B	98 B	99 B	100 B
101 C	102 C	103 C	104 C	105 C	106 C	107 C	108 C	109 C	110 C	111 C	112 C	113 B	114 B	115 B	116 B	117 B	118 B	119 B	120 B
121 C	122 C	123 C	124 C	125 C	126 C	127 B	128 C	129 C	130 C	131 B	132 C	133 B	134 B	135 B	136 B	137 B	138 B	139 C	140 C
141 B	142 C	143 C	144 C	145 B	146 C	147 B	148 B	149 B	150 B	151 B	152 B	153 B	154 B	155 B	156 B	157 B	158 C	159 C	160 C
161 C	162 C	163 C	164 C	165 C	166 C	167 C	168 C	169 C	170 C	171 B	172 B	173 B	174 B	175 C	176 C	177 C	178 C	179 C	180 C
181 B	182 C	183 B	184 B	185 C	186 C	187 B	188 B	189 B	190 B	191 B	192 B	193 C	194 B	195 B	196 C	197 C	198 C	199 C	200 C
201 C	202 B	203 B	204 B	205 B	206 B	207 B	208 B	209 B	210 B	211 B	212 B	213 B	214 B	215 C	216 C	217 C	218 C	219 C	220 C
221 B	222 B	223 B	224 B	225 B	226 B	227 B	228 B	229 B	230 B	231 B	232 B	233 B	234 B	235 C	236 C	237 C	238 C	239 C	240 C
241 B	242 B	243 B	244 B	245 B	246 B	247 B	248 B	249 B	250 B	251 B	252 B	253 B	254 B	255 C	256 C	257 C	258 C	259 C	260 C
261 B	262 B	263 B	264 B	265 B	266 B	267 B	268 B	269 B	270 B	271 B	272 B	273 B	274 B	275 B	276 C	277 B	278 C	279 C	280 B
281 B	282 B	283 B	284 B	285 B	286 B	287 B	288 B	289 B	290 B	291 B	292 B	293 B	294 C	295 C	296 B	297 B	298 B	299 B	300 B
301 B	302 B	303 B	304 C	305 C	306 B	307 B	308 B	309 B	310 B	311 B	312 B	313 B	314 C	315 C	316 C	317 B	318 B	319 B	320 B
321 B	322 B	323 B	324 B	325 B	326 B	327 B	328 B	329 B	330 B	331 B	332 B	333 B	334 B	335 C	336 B	337 B	338 B	339 B	340 B
341 C	342 C	343 C	344 C	345 B	346 B	347 B	348 B	349 B	350 B	351 B	352 B	353 B	354 B	355 B	356 C	357 B	358 B	359 B	360 C
361 B	362 B	363 B	364 B	365 B	366 B	367 B	368 B	369 B	370 B	371 B	372 B	373 B	374 B	375 B	376 B	377 C	378 C	379 C	380 B
381 C	382 B	383 B	384 B	385 B	386 C	387 C	388 B	389 B	390 B	391 B	392 B	393 C	394 C	395 B	396 B	397 B	398 C	399 C	400 C

Note: letters represent clusters

Figure 8

Data Set 1

K = 4 clusters from K-Means

1 A	2 A	3 A	4 A	5 A	6 A	7 A	8 A	9 D	10 C	11 C	12 C	13 C	14 C	15 D	16 C	17 C	18 C	19 C	20 C
21 A	22 A	23 A	24 A	25 A	26 A	27 A	28 A	29 D	30 C	31 C	32 C	33 C	34 C	35 D	36 C	37 C	38 C	39 C	40 C
41 A	42 A	43 A	44 A	45 A	46 A	47 A	48 A	49 A	50 C	51 C	52 C	53 D	54 B	55 B	56 D	57 C	58 D	59 D	60 D
61 A	62 A	63 A	64 A	65 A	66 A	67 A	68 A	69 A	70 D	71 D	72 D	73 D	74 D	75 B	76 B	77 D	78 D	79 D	80 C
81 A	82 A	83 D	84 C	85 C	86 C	87 C	88 C	89 C	90 C	91 C	92 D	93 D	94 B	95 B	96 B	97 B	98 D	99 D	100 B
101 C	102 C	103 C	104 C	105 C	106 C	107 C	108 C	109 C	110 C	111 C	112 D	113 D	114 D	115 B	116 B	117 B	118 D	119 D	120 D
121 C	122 C	123 C	124 C	125 C	126 D	127 D	128 D	129 D	130 C	131 D	132 D	133 B	134 B	135 B	136 B	137 B	138 D	139 D	140 D
141 D	142 D	143 C	144 C	145 D	146 D	147 D	148 D	149 D	150 D	151 D	152 B	153 B	154 D	155 D	156 D	157 D	158 C	159 C	160 C
161 D	162 C	163 C	164 C	165 D	166 C	167 D	168 D	169 D	170 C	171 D	172 B	173 B	174 B	175 C	176 C	177 C	178 C	179 C	180 C
181 D	182 D	183 D	184 D	185 D	186 D	187 D	188 D	189 D	190 D	191 D	192 D	193 D	194 B	195 D	196 C	197 C	198 C	199 C	200 C
201 D	202 D	203 D	204 D	205 D	206 D	207 D	208 B	209 B	210 B	211 D	212 B	213 B	214 B	215 C	216 C	217 C	218 C	219 C	220 C
221 D	222 D	223 D	224 B	225 B	226 D	227 D	228 B	229 B	230 D	231 D	232 D	233 D	234 D	235 C	236 C	237 C	238 C	239 C	240 C
241 B	242 B	243 B	244 B	245 B	246 B	247 B	248 D	249 B	250 B	251 B	252 B	253 B	254 B	255 D	256 C	257 C	258 C	259 C	260 C
261 B	262 B	263 D	264 D	265 B	266 B	267 D	268 D	269 B	270 B	271 B	272 B	273 D	274 B	275 D	276 C	277 B	278 D	279 C	280 D
281 B	282 D	283 D	284 D	285 B	286 B	287 B	288 D	289 B	290 D	291 D	292 D	293 D	294 D	295 D	296 B	297 D	298 B	299 B	300 B
301 D	302 D	303 D	304 D	305 C	306 D	307 B	308 B	309 B	310 B	311 B	312 B	313 D	314 C	315 C	316 D	317 D	318 D	319 D	320 B
321 D	322 D	323 D	324 B	325 D	326 D	327 D	328 B	329 D	330 D	331 B	332 B	333 D	334 D	335 D	336 D	337 B	338 B	339 D	340 D
341 C	342 D	343 D	344 D	345 D	346 D	347 D	348 B	349 B	350 D	351 B	352 D	353 D	354 D	355 D	356 D	357 D	358 D	359 D	360 D
361 D	362 D	363 D	364 B	365 D	366 B	367 D	368 D	369 D	370 B	371 D	372 D	373 B	374 D	375 D	376 B	377 D	378 C	379 C	380 D
381 D	382 D	383 D	384 D	385 D	386 C	387 C	388 D	389 B	390 D	391 D	392 D	393 D	394 C	395 B	396 B	397 D	398 C	399 C	400 C

Note: letters represent clusters

Figure 9

Data Set 1

K = 5 clusters from K-Means

1 A	2 A	3 A	4 A	5 A	6 A	7 A	8 A	9 D	10 C	11 C	12 C	13 C	14 C	15 E	16 C	17 C	18 C	19 C	20 C
21 A	22 A	23 A	24 A	25 A	26 A	27 A	28 A	29 D	30 C	31 C	32 C	33 E	34 E	35 E	36 C	37 C	38 C	39 C	40 E
41 A	42 A	43 A	44 A	45 A	46 A	47 A	48 A	49 A	50 C	51 C	52 C	53 E	54 B	55 B	56 E	57 C	58 E	59 E	60 D
61 A	62 A	63 A	64 A	65 A	66 A	67 A	68 A	69 A	70 D	71 E	72 D	73 D	74 E	75 B	76 B	77 D	78 E	79 D	80 E
81 A	82 A	83 D	84 C	85 C	86 C	87 C	88 C	89 E	90 E	91 E	92 E	93 D	94 B	95 B	96 B	97 B	98 E	99 D	100 B
101 C	102 C	103 C	104 C	105 C	106 C	107 E	108 E	109 C	110 C	111 C	112 E	113 D	114 E	115 B	116 B	117 B	118 D	119 E	120 D
121 E	122 C	123 C	124 C	125 C	126 E	127 D	128 E	129 E	130 E	131 D	132 E	133 B	134 B	135 B	136 B	137 B	138 D	139 E	140 E
141 D	142 E	143 E	144 C	145 E	146 E	147 D	148 D	149 E	150 D	151 E	152 D	153 D	154 D	155 D	156 D	157 E	158 E	159 C	160 C
161 E	162 E	163 E	164 E	165 E	166 E	167 D	168 D	169 D	170 E	171 D	172 B	173 B	174 B	175 C	176 C	177 C	178 C	179 C	180 C
181 D	182 E	183 D	184 D	185 E	186 E	187 E	188 D	189 E	190 E	191 D	192 E	193 E	194 B	195 D	196 C	197 C	198 C	199 C	200 C
201 E	202 D	203 E	204 D	205 E	206 D	207 D	208 B	209 B	210 B	211 E	212 B	213 B	214 B	215 E	216 C	217 C	218 C	219 C	220 C
221 D	222 E	223 E	224 B	225 B	226 D	227 E	228 B	229 B	230 D	231 D	232 D	233 D	234 E	235 C	236 C	237 C	238 C	239 C	240 C
241 B	242 B	243 B	244 B	245 B	246 B	247 B	248 D	249 B	250 B	251 B	252 B	253 B	254 B	255 E	256 C	257 C	258 C	259 C	260 E
261 B	262 B	263 D	264 D	265 B	266 B	267 E	268 D	269 B	270 B	271 B	272 B	273 D	274 D	275 E	276 E	277 B	278 E	279 E	280 E
281 B	282 D	283 E	284 D	285 B	286 B	287 B	288 D	289 B	290 D	291 D	292 D	293 D	294 D	295 E	296 B	297 D	298 B	299 B	300 B
301 E	302 E	303 E	304 E	305 E	306 D	307 B	308 B	309 B	310 B	311 B	312 B	313 E	314 E	315 C	316 E	317 D	318 D	319 D	320 B
321 D	322 E	323 E	324 B	325 D	326 E	327 D	328 B	329 D	330 D	331 B	332 B	333 D	334 D	335 E	336 D	337 B	338 B	339 D	340 D
341 E	342 E	343 E	344 E	345 D	346 E	347 D	348 B	349 B	350 B	351 B	352 D	353 D	354 D	355 D	356 E	357 E	358 D	359 E	360 E
361 D	362 E	363 E	364 B	365 D	366 B	367 D	368 D	369 E	370 B	371 D	372 D	373 B	374 D	375 D	376 B	377 E	378 E	379 E	380 E
381 E	382 D	383 E	384 E	385 E	386 E	387 C	388 E	389 B	390 E	391 D	392 D	393 E	394 C	395 B	396 B	397 E	398 C	399 C	400 C

Note: letters represent clusters

Figure 10

Data Set 1

K = 6 clusters from K-Means

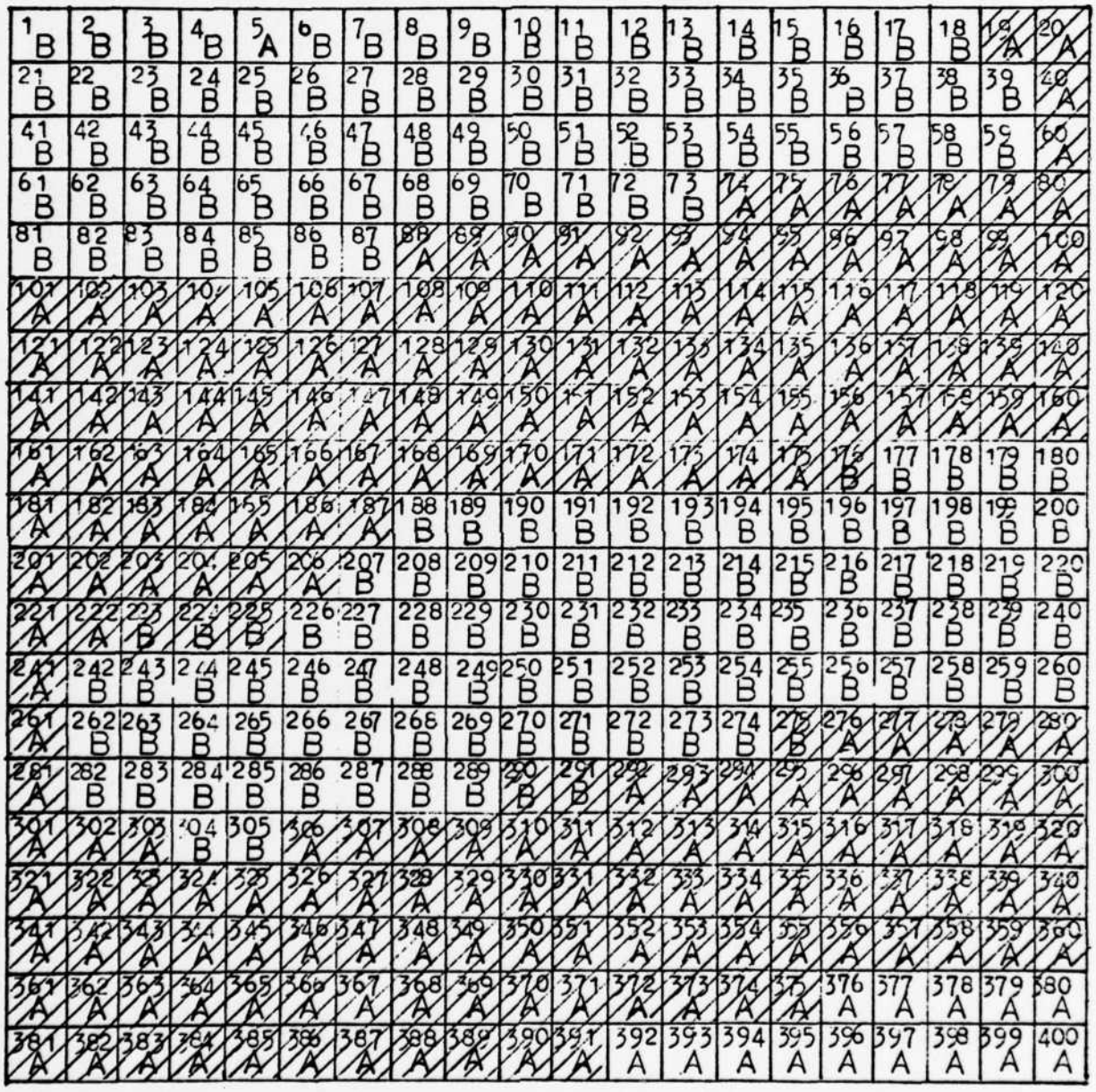
1 A	2 A	3 A	4 A	5 A	6 A	7 A	8 A	9 D	10 C	11 C	12 C	13 C	14 C	15 E	16 C	17 C	18 C	19 C	20 C
21 A	22 A	23 A	24 A	25 A	26 A	27 A	28 A	29 D	30 C	31 C	32 C	33 E	34 E	35 E	36 C	37 C	38 C	39 C	40 E
41 A	42 A	43 A	44 A	45 A	46 A	47 A	48 A	49 A	50 C	51 C	52 C	53 E	54 F	55 F	56 E	57 C	58 E	59 E	60 D
61 A	62 A	63 A	64 A	65 A	66 A	67 A	68 A	69 A	70 D	71 E	72 D	73 D	74 E	75 F	76 F	77 D	78 E	79 D	80 E
81 A	82 A	83 D	84 C	85 C	86 C	87 C	88 C	89 E	90 E	91 E	92 E	93 D	94 F	95 F	96 B	97 F	98 E	99 D	100 F
101 C	102 C	103 C	104 C	105 C	106 C	107 E	108 E	109 C	110 C	111 C	112 E	113 D	114 E	115 F	116 F	117 F	118 D	119 E	120 D
121 E	122 C	123 C	124 C	125 C	126 E	127 D	128 E	129 E	130 E	131 D	132 E	133 B	134 F	135 F	136 B	137 B	138 D	139 E	140 E
141 D	142 E	143 E	144 C	145 E	146 E	147 D	148 D	149 E	150 D	151 E	152 D	153 B	154 D	155 D	156 D	157 E	158 E	159 C	160 C
161 E	162 E	163 E	164 E	165 E	166 E	167 D	168 D	169 D	170 E	171 D	172 B	173 B	174 F	175 C	176 C	177 C	178 C	179 C	180 C
181 D	182 E	183 D	184 D	185 E	186 E	187 E	188 D	189 E	190 E	191 D	192 E	193 E	194 B	195 D	196 C	197 C	198 C	199 C	200 C
201 E	202 D	203 E	204 D	205 E	206 D	207 D	208 F	209 B	210 B	211 E	212 F	213 B	214 F	215 E	216 C	217 C	218 C	219 C	220 C
221 D	222 E	223 E	224 F	225 F	226 D	227 E	228 F	229 B	230 D	231 D	232 D	233 D	234 E	235 C	236 C	237 C	238 C	239 C	240 C
241 B	242 F	243 B	244 B	245 F	246 F	247 B	248 D	249 B	250 F	251 F	252 F	253 F	254 B	255 E	256 C	257 C	258 C	259 C	260 E
261 F	262 F	263 D	264 D	265 F	266 F	267 E	268 D	269 F	270 F	271 F	272 F	273 D	274 D	275 E	276 E	277 F	278 E	279 E	280 E
281 F	282 D	283 E	284 D	285 F	286 F	287 F	288 D	289 F	290 D	291 D	292 D	293 D	294 D	295 E	296 F	297 D	298 B	299 F	300 B
301 E	302 E	303 E	304 E	305 E	306 D	307 F	308 F	309 F	310 F	311 F	312 F	313 E	314 E	315 C	316 E	317 D	318 D	319 D	320 F
321 D	322 E	323 E	324 F	325 D	326 E	327 D	328 F	329 D	330 D	331 F	332 F	333 D	334 D	335 E	336 D	337 F	338 F	339 D	340 D
341 E	342 E	343 E	344 E	345 D	346 E	347 D	348 F	349 F	350 F	351 B	352 D	353 D	354 D	355 D	356 E	357 E	358 D	359 E	360 E
361 D	362 E	363 E	364 F	365 D	366 F	367 D	368 D	369 E	370 F	371 D	372 D	373 F	374 D	375 D	376 F	377 E	378 E	379 E	380 E
381 E	382 D	383 E	384 E	385 E	386 E	387 C	388 E	389 B	390 E	391 D	392 D	393 E	394 C	395 F	396 B	397 E	398 C	399 C	400 C

Note: letters represent clusters

Figure 11

Data Set 2

K = 2 clusters from K-Means

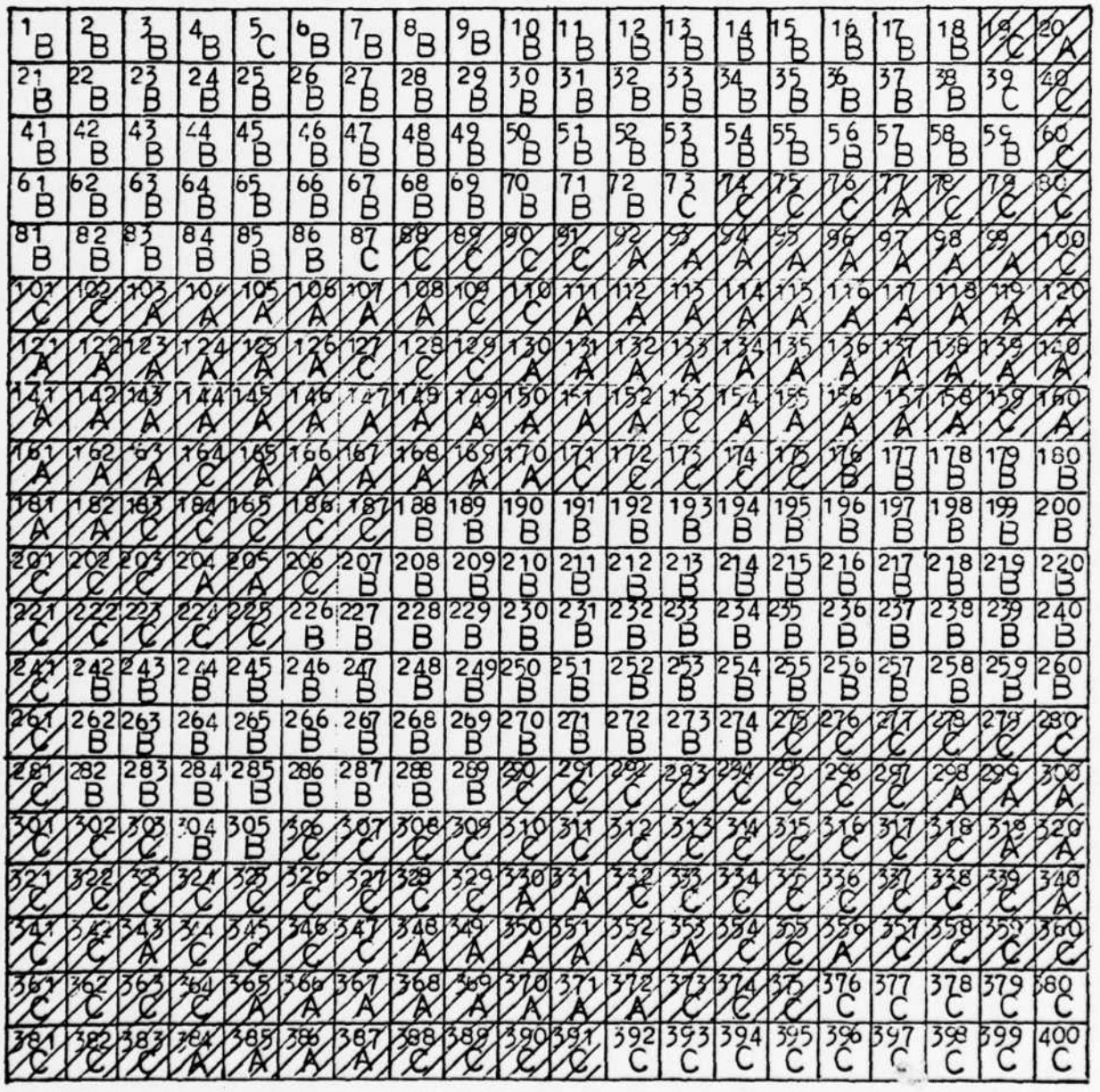


Note: letters represent clusters

Figure 12

Data Set 2

K = 3 clusters from K-Means

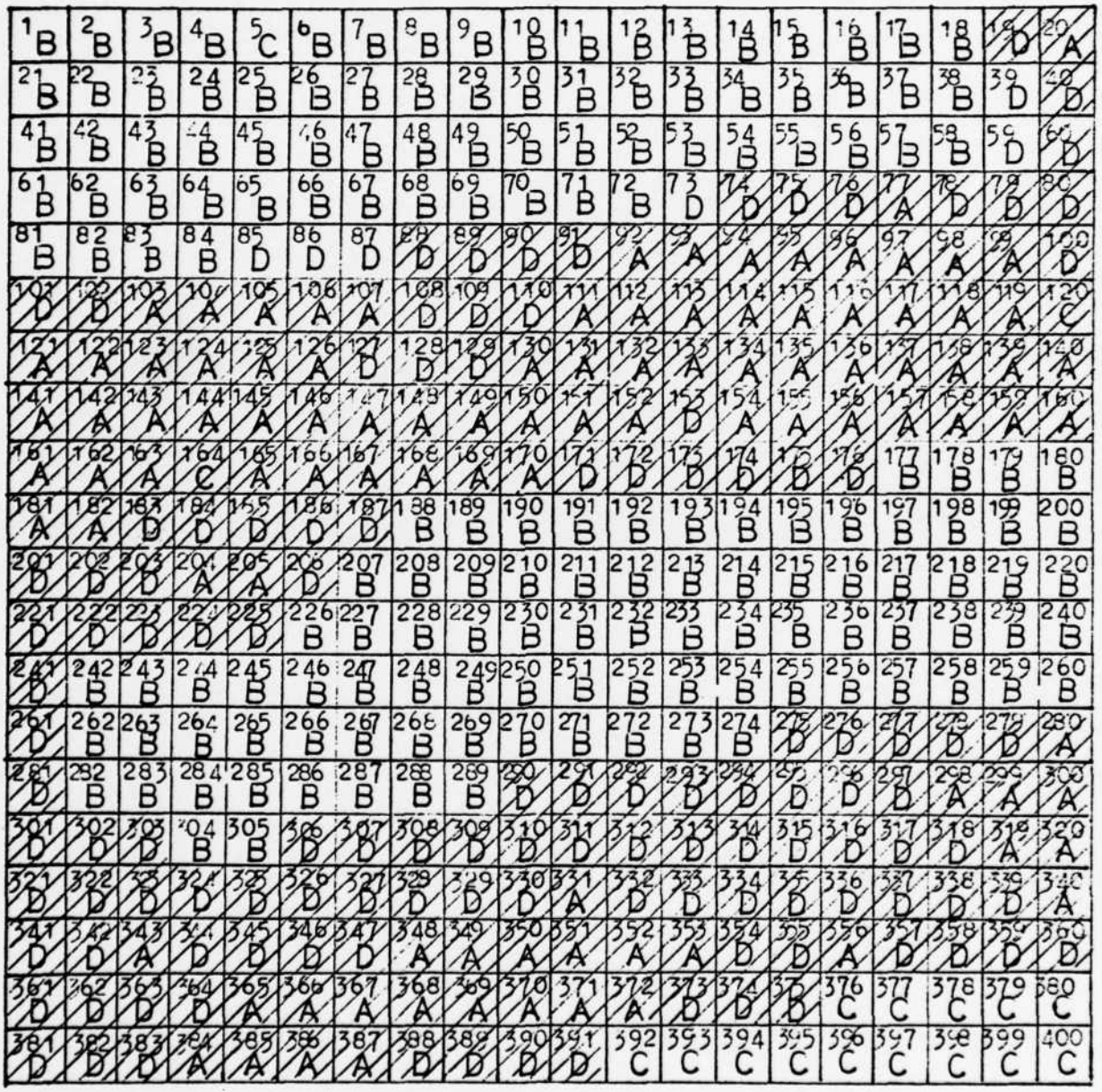


Note: letters represent clusters

Figure 13

Data Set 2

K = 4 clusters from K-Means



Note: letters represent clusters

Figure 14

Data Set 2

K = 5 clusters from K-Means

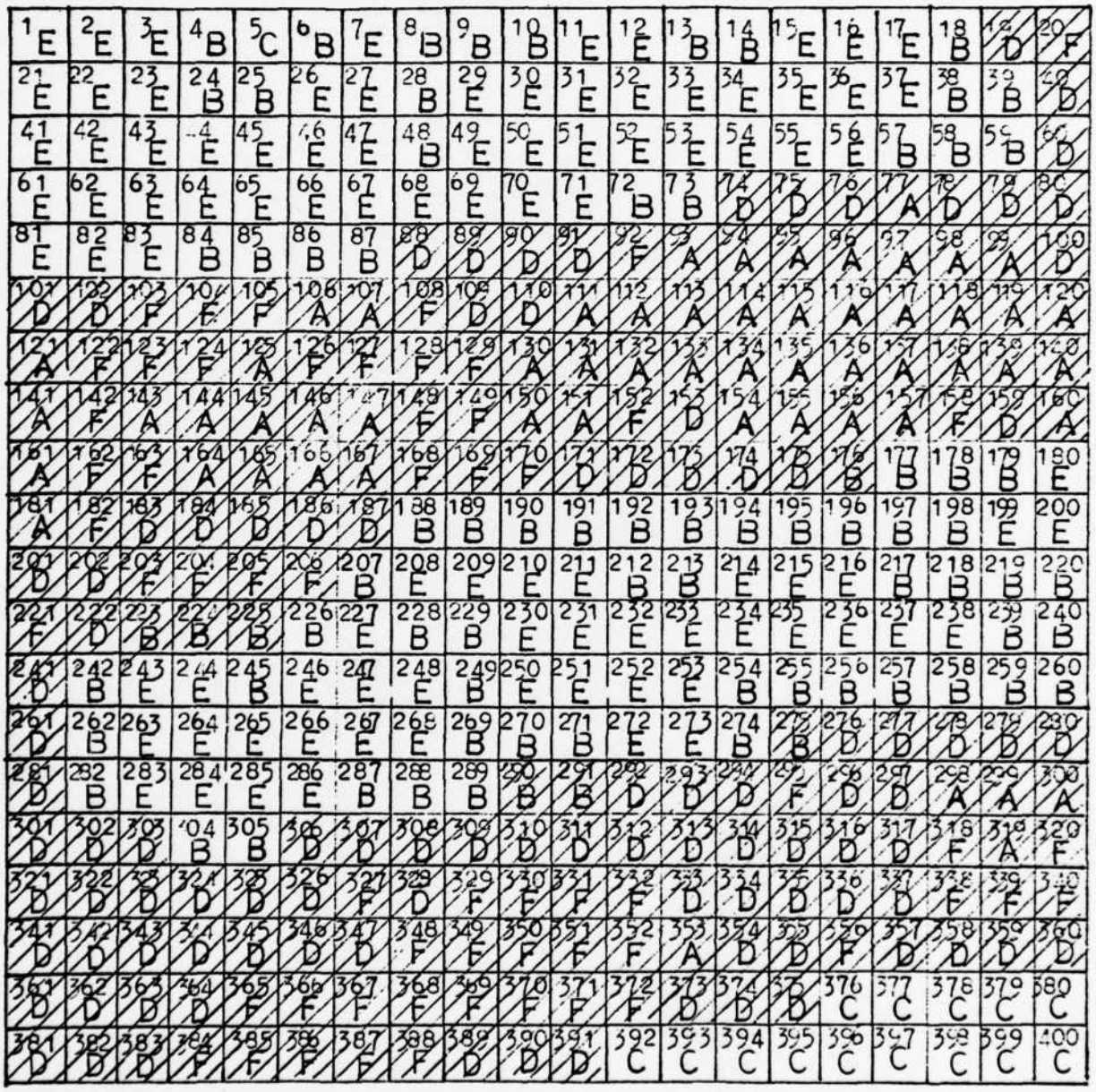
1 E	2 E	3 E	4 B	5 C	6 B	7 E	8 B	9 B	10 B	11 E	12 E	13 B	14 B	15 E	16 E	17 E	18 B	19 D	20 A
21 E	22 E	23 E	24 B	25 B	26 E	27 E	28 B	29 E	30 E	31 E	32 E	33 E	34 E	35 E	36 E	37 E	38 B	39 B	40 D
41 E	42 E	43 E	44 E	45 E	46 E	47 E	48 B	49 E	50 E	51 E	52 E	53 E	54 E	55 E	56 E	57 B	58 B	59 B	60 D
61 E	62 E	63 E	64 E	65 E	66 E	67 E	68 E	69 E	70 E	71 E	72 B	73 B	74 D	75 D	76 D	77 A	78 D	79 D	80 D
81 E	82 E	83 E	84 B	85 B	86 B	87 B	88 D	89 D	90 D	91 D	92 A	93 A	94 A	95 A	96 A	97 A	98 A	99 A	100 D
101 D	102 D	103 D	104 A	105 A	106 A	107 A	108 D	109 D	110 D	111 A	112 A	113 A	114 A	115 A	116 A	117 A	118 A	119 A	120 C
121 A	122 A	123 A	124 A	125 A	126 A	127 D	128 D	129 D	130 A	131 A	132 A	133 A	134 A	135 A	136 A	137 A	138 A	139 A	140 A
141 A	142 A	143 A	144 A	145 A	146 A	147 A	148 A	149 A	150 A	151 A	152 A	153 D	154 A	155 A	156 A	157 A	158 A	159 D	160 A
161 A	162 A	163 A	164 C	165 A	166 A	167 A	168 A	169 A	170 A	171 D	172 D	173 D	174 D	175 D	176 B	177 B	178 B	179 B	180 E
181 A	182 A	183 D	184 D	185 D	186 D	187 D	188 B	189 B	190 B	191 B	192 B	193 B	194 B	195 B	196 B	197 B	198 B	199 E	200 E
201 D	202 D	203 D	204 A	205 A	206 D	207 B	208 E	209 E	210 E	211 E	212 B	213 B	214 E	215 E	216 E	217 B	218 B	219 B	220 B
221 D	222 D	223 B	224 B	225 B	226 B	227 E	228 B	229 B	230 E	231 E	232 E	233 E	234 E	235 E	236 E	237 E	238 E	239 B	240 B
241 D	242 B	243 E	244 E	245 B	246 E	247 E	248 E	249 B	250 E	251 E	252 E	253 E	254 B	255 B	256 B	257 B	258 B	259 B	260 B
261 D	262 B	263 E	264 E	265 E	266 E	267 E	268 E	269 B	270 B	271 B	272 E	273 E	274 B	275 B	276 D	277 D	278 D	279 D	280 D
281 D	282 B	283 E	284 E	285 E	286 E	287 B	288 B	289 B	290 B	291 B	292 D	293 D	294 D	295 D	296 D	297 D	298 A	299 A	300 A
301 D	302 D	303 D	304 B	305 B	306 D	307 D	308 D	309 D	310 D	311 D	312 D	313 D	314 D	315 D	316 D	317 D	318 D	319 A	320 A
321 D	322 D	323 D	324 D	325 D	326 D	327 D	328 D	329 D	330 D	331 A	332 D	333 D	334 D	335 D	336 D	337 D	338 D	339 D	340 A
341 D	342 D	343 D	344 D	345 D	346 D	347 D	348 A	349 A	350 A	351 A	352 A	353 A	354 D	355 D	356 D	357 D	358 D	359 D	360 D
361 D	362 D	363 D	364 D	365 A	366 A	367 A	368 A	369 A	370 A	371 A	372 A	373 D	374 D	375 D	376 C	377 C	378 C	379 C	380 C
381 D	382 D	383 D	384 A	385 A	386 A	387 A	388 D	389 D	390 D	391 D	392 C	393 C	394 C	395 C	396 C	397 C	398 C	399 C	400 C

Note: letters represent clusters

Figure 15

Data Set 2

K = 6 clusters from K-Means



Note: letters represent clusters

classifications. Pixels 5, 223-225, 275, 290-291, 376-389, and 392-400, which were mismatched in Figure 11, are all in cluster C. This implies that cluster C is a shade of gray between light and dark (a borderline shade between clusters A and B).

For K = 4 in Figure 13, all the borderline pixels (176, 223-225, 275, 290-291) were in Cluster D which appears to be the borderline shade of gray. Pixels 5, 376-380, and 392-400 are all in Cluster C, which appears to be a light shade different from the red area as seen in Figure 1.

For K = 5 and 6 in Figures 14 and 15, Cluster C remains unchanged, while Clusters B and D appear to represent borderline shades of gray that are between shades for Cluster 1 (light) and Cluster 5 (dark). Pixels 176, 223-225, 275, and 290-291 have all changed to Cluster B from Cluster D. This is an even stronger indication of the borderline nature of these pixels. Cluster F is a further breakout of light shades from Clusters A and D.

For all K, the clusters follow the general hatched area structure extremely well.

5.3 Problems: The K-Means method has several problems as identified above. First, a major problem is not knowing what K is. Second, there are assumptions implicit in the method, e.g., cluster structures which are spherical. Third, the method guarantees only locally, not globally, optimum solutions which depend on scaling and initial partitioning configurations. Finally, little is known about when a global optimum has been found.

The results of Section 5.2 highlight the problem of choosing K. In the actual application, the K-Means program was run for K = 2 to K = 10 on both data sets. In both

sets, the results were not helpful in choosing K.

Also, geographical information was not considered in forming the clusters. For agricultural sites involving fields of specific crops, this results in lost information otherwise available when pixel location is taken into account.

Chapter 6

The Kth Nearest Neighbor Clustering Method

6.0 Introduction: This chapter describes in detail the relatively new K^{th} nearest neighbor clustering method developed by Wong and Lane (1983), presents a literature survey on similar clustering procedures and their limitations, provides the results of applying this method to the two data sets for various values of K, and discusses the method's strengths and weaknesses. Much of the material in the first part of this chapter is taken from Wong and Lane (1983) and Wong and Schaack (1982).

6.1 Method Description: In this section, motivation will be given for the method, followed by a literature survey, and finally a detailed description of the entire clustering procedure.

6.1.1 Background: A recent study of Blashfield and Aldenderfer (1978) shows that numerous clustering methods have been developed in the past two decades. A review of many of these techniques can be found in Cormack (1971), Anderberg (1973), Sneath and Sokal (1973), Everitt (1974), Hartigan (1975), and Spath (1980). The validity of the sample clusters obtained by these methods is questionable, however, due to the lack of development within a theoretical framework of the probabilistic and statistical aspects of clustering methodology and research. Consequently, the existing clustering procedures are often regarded as heuristics generating artificial clusters from a given set of sample data. Similar to the K-Means method, the task of determining the number of clusters (subpopulations) from the data is a major problem of this procedure. As a result, Wong and Lane (1983) and Wong and Schaach (1982)

saw the need for a cluster procedure that was useful for drawing statistical inferences about the underlying population and determining the number of clusters (subpopulations) from the sample data.

To this end, they developed a clustering procedure based on the K^{th} nearest neighbor density estimate, which they show to be set-consistent (defined below) for high-density clusters in several dimensions. Further, they showed that the procedure is useful in providing : (1) a diagnostic plot which helps to indicate the number of subpopulations (modes) present, and (2) a bootstrap procedure for testing the existence of two or more subpopulations. Although the bootstrap procedure is well defined for multivariate data such as the Landsat data, the probability cutoff values for their sample test statistic cannot be easily generalized to several dimensions. For this reason, the bootstrap procedure will not be used in this study.

6.1.2 A Theoretical Approach to Forming Clusters: As previously mentioned, the K^{th} nearest neighbor density estimate and clustering procedure is set-consistent for high-density clusters in several dimensions. The set-consistency property of hierarchial clustering procedures will be rigorously defined in this section.

To be able to evaluate sampling properties of a cluster procedure, the population clusters must be defined on a population density function under which the sample was taken. Let sample observations $X_1, X_2, \ldots, X_N$ in P-dimensional space be taken from a density function f, taken with respect to Lebesgue measure (see Wong and Lane (1983). Then, using Hartigan's high-density clustering model (1975, p. 205), define the true population clusters on f as high-density clusters. In essence, a high-density

cluster at level f_0 in the population f is defined as a maximal connected set $\{x|f(x) \geq f_0\}$. The family T of all such clusters forms a tree, since if $A \in T$, $B \in T$ then either $A \supset B$, $B \supset A$, or $A \cap B = \emptyset$.

Any hierarchical (joining) clustering procedure which results in a sample cluster tree T_N on the observations $X_1, X_2, \ldots, X_N$ can be evaluated to see if T_N converges to T with probability 1 as N approaches infinity. Such a method (T_N) is said to be strongly set-consistent for high-density clusters (T) if for all $A, B \in T$, and $A \cap B = \emptyset$, Prob $(A_N \cap B_N = \emptyset$ as $N \to \infty) = 1$, where A_N and B_N are the smallest clusters in T_N cpntaining all sample points in A and B. Since $A \subset B$ implies $A_N \subset B_N$, the limiting result implies that the tree relationship T_N converges strongly to T. This consistent cluster-estimation problem under the density-contour clustering model has been addressed by Hartigan (1981) and Wong and Lane (1983).

With this definition of consistency, hierarchical clustering methods can be evaluated by examining high-density clusters for strong set-consistency. If a method is set-consistent, the sequence of larger and larger hierarchical clusters it generates for a sample are sets of observations lying in decreasing density contours of the underlying distribution. These sample high-density clusters help indicate the number and location of modal regions in the population space. Unlike many other clustering methods which impose structure (explicit or implied) on their clusters (e.g., K-Means assumes spherical clusters), the geometrical shape of the population density contours determines how the sample high-density clusters are configured. Hence, a set-consistent clustering procedure does not predetermine any specific structure on its clusters. (See also Everitt, 1974, Chapter 4, for some other well-known

methods, that assume a spherical shape for their clusters). For the opposite situation, clustering procedures that are not set-consistent do not adapt to the underlying distribution and are not suitable for detecting high-density or "natural" clusters (see Carmichael (et. al. 1968).

Under the high-density clustering model, density estimates are used to produce sample clusters, i.e., the high-density clusters from the estimates. Any clustering procedure is expected to be set-consistent for high-density clusters of the procedure is based on a density estimate that is uniformly consistent. Set-consistency is lacking in present clustering procedures.

6.1.3 Literature Survey: Hartigan (1977a, 1977b, 1979, 1981) has examined the set-consistency of many of the best known hierarchical high-density clustering methods. He has shown (1981) that most are not set-consistent. The complete linkage (Sorenson 1948) and average linkage (Sneath and Sokal 1973) methods are not set-consistent, while single linkage (Sneath 1957) is weakly set-consistent in one dimension only. As these results show, most of the evaluative work on the high-density clustering model is complete. However, little work had been done on developing set-consistent clustering procedures for high-density clusters. Wong (1982) developed a hybrid clustering method which is weakly set-consistent in one dimension and probably in several dimensions. Although this hybrid method is practicable for large data sets, it is ill-suited for small samples ($n < 100$), and can only be applied to case-by-variable matrices. Wong and Lane (1983) developed a strongly set-consistent high-density clustering procedure which is suited to smaller data sets and to both case-by-variable data matrices and case-by-case distance matrices.

Wishart (1969) developed the "Mode Analysis" procedure which is related to the K^{th} nearest neighbor method. While attempting to improve on the single linkage clustering technique, his procedure was not designed to extract high-density clusters from the density estimate. Therefore, its set-consistency for high-density clusters was never established.

Single linkage corresponds to estimating a density by the nearest neighbor method (Hartigan 1977b), where the density estimate $f_N(X)$ at a point X is inversely proportional to the volume of the smallest closed sphere containing one other sample point. This estimate is not consistent in the sense that $f_N(X)$ does not approach $f(X)$ in probability. The K^{th} nearest neighbor method provides an improved density estimate and probably improved clusters as well. Here the estimated density at a point X is $f_N(X) = K/(NV_K(X))$, where $V_K(X)$ is the volume of the sphere centered at X containing K additional sample points. This estimate is uniformly consistent with probability 1 if f is uniformly continuous and if $K = K(N)$ satisfies $K(N)/N \to 0$ and $K(N)/\log N \to \infty$ as $N \to \infty$. (See Wong and Lane 1983).

Several authors have also studied the problem of testing for clusters. Engelman and Hartigan (1969) and Hartigan (1978) employ a likelihood ratio approach to testing the presence of one or two univariate normal populations. Lee (1979) provides a multivariate generalization of their work. He developed a multivariate test for clusters based on the union-intersection principle of test construction. The major deficiency of these tests is that they are based on a restrictive parametric clustering model, where clusters are formed by normal density mixtures. On the other hand, Wong and Schaack (1983) used a nonparametric density-contour

clustering model, where the density contours of the underlying density function define the clusters. They developed procedures that are useful for assessing and testing multimodality. Their capability for assessment will be applied to the two data sets of this study.

6.1.4 The K^{th} Nearest Neighbor Clustering Algorithm: This section describes the clustering procedure of Wong and Lane (1983) for deriving the tree of sample high-density clusters from the K^{th} nearest neighbor density estimate. Their procedure consists of two parts: density estimation and hierarchical clustering. The K^{th} nearest neighbor density estimation algorithm is first completed in order to provide a uniformly consistent density estimate. Then, the clustering algorithm is invoked to obtain the tree of sample high-density clusters on the estimated underlying density.

6.4.1 The Density Estimation Algorithm: In the first phase of their procedure, the K^{th} nearest neighbor density estimation algorithm is used to obtain a strongly uniform, consistent estimate of the underlying density. This K^{th} nearest neighbor estimate $f_N(X)$ of the density f at a point X is $f_N(X) = K/(NV_K(X))$, where $V_K(X)$ is the volume of the smallest sphere centered at X and containing at least K of the independent, identically distributed (i.i.d.) random vectors $X_1, X_2, \ldots, X_N$ with values in $\mathbb{R}^P$, $P \geq 1$, from the common probability density f. The proof that this estimate is strongly uniform consistent is based on the lemma in Devroye and Wagner (1977):

> If f is uniformly continuous on $\mathbb{R}^P$ and if $K = K(N)$ is a sequence of positive integers satisfying:
>
> (a) $K(N)/N \to 0$, and
>
> (b) $K(N)/\log N \to \infty$, as $N \to \infty$,

then $\sup_X |f_N(X) - f(X)| \to 0$ with probability 1.

Once this density estimate is obtained, their procedure searches for the population high-density clusters given the random sample from the underlying distribution F with density f. To this end, the high-density clusters defined on the density estimate f_N are used as sample estimates of the population clusters. These hierarchical sample clusters are constructed in the second part of their procedure.

6.1.4.2 The Hierarchical Clustering Algorithm: The second part of their clustering procedure consists of constructing the high-density clusters. Initially, a distance matrix is computed between all neighboring points. Two points are considered neighbors if at least one is one of the K^{th} nearest neighbors of the other. The distance between neighbors is defined to be inversely proportional to a pooled density estimate at the point halfway between them. Once this initial stage is complete, a single linkage clustering algorithm (see Sneath, 1957) is applied to the the distance matrix to form the tree of sample high-density clusters.

A more rigorous description of the distance matrix will now be given. It requires two preliminary definitions:

Definition 1: Two observations X_i and X_j are "neighbors" if $d^*(X_i,X_j) \leq d_K(X_i)$ or $d_K(X_j)$, where d^* is the Euclidean metric and $d_K(X_i)$ is the K^{th} nearest neighbor distance to X_i.

Definition 2: The distance $D(\cdot,\cdot)$ between observations X_i and X_j is $D(X_i,X_j) = (1/2)[1/f_N(X_i) + 1/f_N(X_j)] = (N/2K)[V_K(X_i) + V_K(X_j)]$ if X_i and X_j are neighbors, and $D(X_i,X_j) = \infty$, otherwise.

Thus, the distance matrix $D(X_i,X_j)$ for $1 \leq i, j \leq N$ is defined for the N vector observations. Finite distances are defined only for pairs of observations in the same neighborhood in $\mathbb{R}^P$, and these distances are inversely proportional to the pooled estimated density at the midpoint between the pair.

After the distance matrix D is computed, a single linkage clustering algorithm is applied to D to form the sample high-density cluster tree. This clustering algorithm forms single linkage clusters from a set of observations $X_1, X_2, \ldots, X_N$ with distances $D(X_i,X_j)$ for $1 \leq i, j \leq N$ in the distance matrix D. This algorithm can be described as follows:

(a) Let X_i and X_j be the closest pair of points (smallest $D(X_i,X_j)$).

(b) Amalgamate them to form cluster C_1 and define the distance $D(C_1,X_m)$ between the cluster and any observation X_m as $D(C_1,X_m) = \min [D(X_i,X_m), D(X_j,X_m)]$.

(c) Repeat steps (a) and (b) treating cluster C_1 as an observation and removing X_i, X_j from further consideration.

(d) This amalgamation process stops when there is only one large cluster remaining.

All clusters formed by this hierarchical process are called single linkage clusters. Gower and Ross (1969), and Hartigan (1975) describe computational single linkage algorithms. At each step in the above process, an amalgamated cluster is a "maximal linked" set, where X_i and X_j are said to be <u>linked</u> whenever $D(X_i,X_j)$ is $\leq$ distance D_0. Since the distance between neighbors is the reciprocal of the density estimate f_N at the point midway between them, every cluster formed has the property

that the density estimates over the observations in this cluster are $\geq$ a specific density level f_0. Then, since the distance between two observations is only defined for neighboring pairs of observations, all formed single linkage clusters are maximal connected sets of the form $\{X/f_N(X) \geq f_0\}$. These sets are the high-denisty clusters on f_N.

6.1.4.3 The Computational Algorithm: Since the high-density clusters constructed above are invariant to monotone transformations of the density f, the K^{th} nearest neighbor distances $d_K(X_i)$ for $i = 1$ to N can be used in place of the volumes $V_K(X_i)$. Then, the K^{th} nearest neighbor clustering procedure can be accomplished using the following computational algorithm.

(a) Compute $d_K(X_i)$ for $i = 1$ to N, where $c_K(X_i)$ is the K^{th} nearest neighbor distance of X_i. Friedman et. al. (1975) presents a computationally efficient algorithm for finding these distances.

(b) Compute the distance matrix D of the $D(X_i,X_j)$'s by:
$D(X_i,X_j) = (1/2)[d_K(X_i) + d_K(X_j)]$ if
$d^*(X_i,X_j) \leq d_K(X_i)$ or $d^*(X_i,X_j) \leq d_K(X_j)$, where
$d^*(X_i,X_j)$ is the Euclidean metric;
Otherwise, $D(X_i,X_j) = \infty$.

(c) Once D is computed, apply to it the single linkage clustering algorithm to form the sample tree of high-density clusters.

The computational requirements for this algorithm (see Wong and Lane, 1983) are greater than those of the hybrid clustering method of Wong (1982). For this reason it is not practicable for large data sets. However, it is well suited to small samples and can be applied to both

case-by-variable data and case-by-case dissimilarity matrices.

6.1.4.4 Strong Set-Consistency of K^{th} Nearest Neighbor Clustering (see Wong and Lane, 1983): The asymptotic consistence of the K^{th} nearest neighbor clustering method for high-density clusters in $\mathbb{R}^P$, $P \geq 1$, stems from the following theorem:

Theorem: Let f be a positive, uniformly continuous function on $\mathbb{R}^P$ such that $\{X/f(X) \geq f_0\}$ is the union of a finite number of compact subsets of $\mathbb{R}^P$ for every $f_0 > 0$. Let T be the tree of high-density clusters defined on f. Suppose A and B are any two disjoint high-density clusters in T with connected interiors. Let $X_1, X_2, \ldots, X_N$ be a random sample from f, and let T be the hierarchical clusters specified by the K^{th} nearest neighbor clustering algorithm.

Provided $K = K(N)$ satisfies (a) $K(N)/N \to 0$ and (b) $K(N)/\log N \to \infty$ as $N \to \infty$,
Then there exist $A_N, B_N \in T_N$ with $A_N \supset A \cap \{X_1, X_2, \ldots, X_N\}$, $B_N \supset B \cap \{X_1, X_2, \ldots, X_N\}$ and $A_N \cap B_N = \emptyset$ with probability 1. And the proof is given in Wong and Lane (1983).

6.1.5 Diagnostic Plots for the Number of Modes: As mentioned in Section 6.1.4, the K^{th} nearest neighbor clustering procedure is strongly consistent only when $K = K(N)$ satisfies $K(N)/N \to 0$ and $K(N)/\log N \to \infty$ as $N \to \infty$. However, in practical situations, the choice of K presents a real problem, since it has not been adequately dealt with in practice or theory. Instead, it is generally suggested to try a range of values of K (see Wong and Schaack, 1982). Wong and Schaack (1982) propose that the number of modes identified in the sample hierarchical

clustering for different values of K should be plotted against K. They claim that such a plot is useful in suggesting the number of modes in the population and give several illustrative examples.

Actually, the value of K controls the amount by which the data are smoothed to obtain the density estimate. In fact, the density estimate becomes less bumpy (fewer modes) when larger values of K are selected, i.e., the number of identified modes is a non-increasing function of K. (The proof is similar to the one given in Silverman (1981) for the kernel density estimate.) For example, if the data are strongly bimodal, a large value of K will be needed to obtain a unimodal estimate. As stated in Wong and Schaack (1982), the plot of the number of estimated modes against K will show a non-increasing step function. They expect that when the number of estimated modes reaches the true number of modes, it will be stable over a range of values of K. Their examples bear this out, and they performed a Monte Carlo study which showed the effectiveness of the diagnostic plot. Therefore, the plot can be used to assess multimodality. Then instead of some arbitrary choice of scale parameters used by most existing methods, the K^{th} nearest neighbor clusltering procedure has the virture of making the choice in a rather automatic, natural way.

Wong and Schaack (1982) not only show the proposed diagnostic plot to be useful in indicating the number of modes present in a given population, but also that it is useful for surfacing the possible existence of even finer subpopulations. They warn that it is sensitive to sample sizes from different populations, but only in as much as they impose upper bounds on the interval of stability. All things considered, their proposed plot appears to be

a practical and valuable diagnostic tool for determining multimodality.

As previously mentioned, they further study a test statistic for testing hypotheses about the number of modes present in a population. However, even though their test statistic is well defined for multivariate data (such as the data in Data Set 1 and Data Set 2), critical values for it cannot be easily generated in several dimensions. Much work still needs to be done in this area to generalize their testing procedure to several dimensions.

6.2 Applying the K^{th} Nearest Neighbor Method: In this section, the K^{th} nearest neighbor clustering procedure described above is applied to the same two data sets to solve the constrained clustering problem. As with the K-Means method, the goal remains to identify contiguous batches of pixels with similar multispectral measurements. Once this is done for several values of K, the proposed diagnostic plot of the estimated number of modes against K will be examined for each data set to determine the number of modes they contain.

Using the basic algorithm of section 6.1.4.3, the density estimates for each set of 400 4-dimensional observations were obtained. Following these estimates, the tree of sample high-density clusters was obtained for several values of K.

6.2.1 Data Set 1 Results: Figures 16 through 21 show these constrained clusters for Data Set 1 after applying the K^{th} nearest neighbor clustering procedure for selected values of K (K = 10, 15, 20, 25, 30, and 40 respectively).

A comparison will now be made between Figures 1 and 2 and Figures 16 through 21 to get a qualitative feel for how

Figure 16

Data Set 1

K^{th} Nearest Neighbor

K = 10

1 A	2 A	3 A	4 A	5 A	6 A	7 A	8 A	9 D	10 B	11 B	12 B	13 B	14 B	15 B	16 B	17 B	18 B	19 B	20 B
21 A	22 A	23 A	24 A	25 A	26 A	27 A	28 A	29 D	30 B	31 B	32 B	33 B	34 B	35 B	36 B	37 B	38 B	39 B	40 B
41 A	42 A	43 A	44 A	45 A	46 A	47 A	48 A	49 A	50 D	51 D	52 D	53 D	54 D	55 C	56 B	57 B	58 B	59 B	60 D
61 A	62 A	63 A	64 A	65 A	66 A	67 A	68 A	69 A	70 D	71 B	72 D	73 D	74 B	75 C	76 C	77 D	78 B	79 D	80 B
81 A	82 A	83 D	84 B	85 B	86 B	87 B	88 B	89 B	90 B	91 B	92 B	93 D	94 C	95 C	96 C	97 C	98 B	99 D	100 C
101 B	102 B	103 B	104 B	105 B	106 B	107 B	108 B	109 B	110 B	111 B	112 B	113 D	114 B	115 D	116 C	117 D	118 D	119 B	120 D
121 B	122 B	123 B	124 B	125 B	126 B	127 D	128 B	129 B	130 B	131 D	132 B	133 C	134 C	135 C	136 C	137 C	138 D	139 B	140 B
141 D	142 B	143 B	144 B	145 B	146 B	147 D	148 D	149 B	150 D	151 B	152 D	153 D	154 D	155 D	156 D	157 B	158 B	159 B	160 B
161 B	162 B	163 B	164 B	165 B	166 B	167 D	168 D	169 D	170 B	171 D	172 C	173 C	174 C	175 B	176 B	177 B	178 B	179 B	180 B
181 D	182 B	183 D	184 D	185 B	186 B	187 B	188 D	189 B	190 B	191 D	192 B	193 B	194 C	195 D	196 B	197 B	198 B	199 B	200 B
201 B	202 D	203 B	204 D	205 B	206 D	207 D	208 C	209 C	210 C	211 B	212 C	213 C	214 C	215 B	216 B	217 B	218 B	219 B	220 B
221 D	222 B	223 B	224 D	225 D	226 D	227 B	228 D	229 D	230 D	231 D	232 D	233 D	234 B	235 B	236 B	237 B	238 B	239 B	240 B
241 C	242 C	243 C	244 C	245 C	246 C	247 C	248 D	249 C	250 C	251 C	252 C	253 C	254 C	255 B	256 B	257 B	258 B	259 B	260 B
261 C	262 C	263 D	264 D	265 C	266 C	267 B	268 D	269 C	270 C	271 C	272 C	273 D	274 D	275 B	276 B	277 C	278 B	279 B	280 B
281 C	282 D	283 B	284 D	285 C	286 B	287 C	288 D	289 C	290 D	291 D	292 D	293 D	294 D	295 B	296 C	297 D	298 C	299 C	300 C
301 B	302 B	303 B	304 B	305 B	306 D	307 C	308 C	309 C	310 C	311 C	312 C	313 B	314 B	315 B	316 B	317 D	318 D	319 D	320 C
321 D	322 B	323 B	324 C	325 D	326 B	327 D	328 C	329 D	330 D	331 C	332 C	333 D	334 D	335 B	336 D	337 C	338 C	339 D	340 D
341 B	342 B	343 B	344 B	345 D	346 B	347 D	348 D	349 D	350 B	351 D	352 D	353 D	354 D	355 D	356 B	357 B	358 D	359 B	360 B
361 D	362 B	363 B	364 C	365 D	366 C	367 D	368 D	369 B	370 C	371 D	372 D	373 C	374 D	375 D	376 C	377 B	378 B	379 B	380 B
381 B	382 D	383 B	384 B	385 B	386 B	387 B	388 B	389 C	390 B	391 D	392 D	393 B	394 B	395 C	396 C	397 B	398 B	399 B	400 B

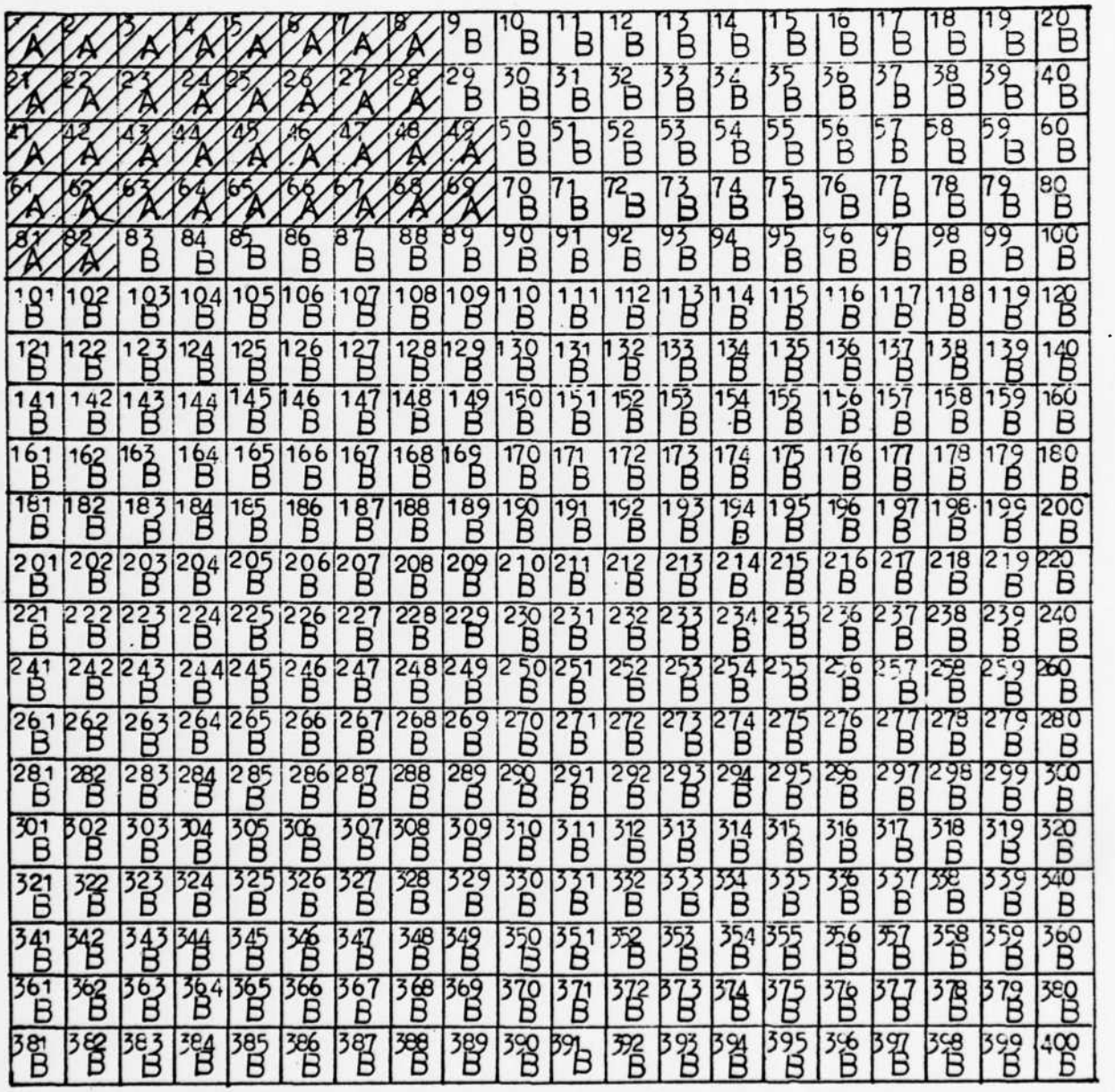
Note: letters represent clusters

Figure 17

Data Set 1

K^{th} Nearest Neighbor

K = 15



Note: letters represent clusters

Figure 18

Data Set 1

K^{th} Nearest Neighbor

K = 20

1 A	2 A	3 A	4 A	5 A	6 A	7 A	8 A	9 B	10 B	11 B	12 B	13 B	14 B	15 B	16 B	17 B	18 B	19 B	20 B
21 A	22 A	23 A	24 A	25 A	26 A	27 A	28 A	29 B	30 B	31 B	32 B	33 B	34 B	35 B	36 B	37 B	38 B	39 B	40 B
41 A	42 A	43 A	44 A	45 A	46 A	47 A	48 A	49 A	50 B	51 B	52 B	53 B	54 B	55 B	56 B	57 B	58 B	59 B	60 B
61 A	62 A	63 A	64 A	65 A	66 A	67 A	68 A	69 A	70 B	71 B	72 B	73 B	74 B	75 B	76 B	77 B	78 B	79 B	80 B
81 B	82 A	83 B	84 B	85 B	86 B	87 B	88 B	89 B	90 B	91 B	92 B	93 B	94 B	95 B	96 B	97 B	98 B	99 B	100 B
101 B	102 B	103 B	104 B	105 B	106 B	107 B	108 B	109 B	110 B	111 B	112 B	113 B	114 B	115 B	116 B	117 B	118 B	119 B	120 B
121 B	122 B	123 B	124 B	125 B	126 B	127 B	128 B	129 B	130 B	131 B	132 B	133 B	134 B	135 B	136 B	137 B	138 B	139 B	140 B
141 B	142 B	143 B	144 B	145 B	146 B	147 B	148 B	149 B	150 B	151 B	152 B	153 B	154 B	155 B	156 B	157 B	158 B	159 B	160 B
161 B	162 B	163 B	164 B	165 B	166 B	167 B	168 B	169 B	170 B	171 B	172 B	173 B	174 B	175 B	176 B	177 B	178 B	179 B	180 B
181 B	182 B	183 B	184 B	185 B	186 B	187 B	188 B	189 B	190 B	191 B	192 B	193 B	194 B	195 B	196 B	197 B	198 B	199 B	200 B
201 B	202 B	203 B	204 B	205 B	206 B	207 B	208 B	209 B	210 B	211 B	212 B	213 B	214 B	215 B	216 B	217 B	218 B	219 B	220 B
221 B	222 B	223 B	224 B	225 B	226 B	227 B	228 B	229 B	230 B	231 B	232 B	233 B	234 B	235 B	236 B	237 B	238 B	239 B	240 B
241 B	242 B	243 B	244 B	245 B	246 B	247 B	248 B	249 B	250 B	251 B	252 B	253 B	254 B	255 B	256 B	257 B	258 B	259 B	260 B
261 B	262 B	263 B	264 B	265 B	266 B	267 B	268 B	269 B	270 B	271 B	272 B	273 B	274 B	275 B	276 B	277 B	278 B	279 B	280 B
281 B	282 B	283 B	284 B	285 B	286 B	287 B	288 B	289 B	290 B	291 B	292 B	293 B	294 B	295 B	296 B	297 B	298 B	299 B	300 B
301 B	302 B	303 B	304 B	305 B	306 B	307 B	308 B	309 B	310 B	311 B	312 B	313 B	314 B	315 B	316 B	317 B	318 B	319 B	320 B
321 B	322 B	323 B	324 B	325 B	326 B	327 B	328 B	329 B	330 B	331 B	332 B	333 B	334 B	335 B	336 B	337 B	338 B	339 B	340 B
341 B	342 B	343 B	344 B	345 B	346 B	347 B	348 B	349 B	350 B	351 B	352 B	353 B	354 B	355 B	356 B	357 B	358 B	359 B	360 B
361 B	362 B	363 B	364 B	365 B	366 B	367 B	368 B	369 B	370 B	371 B	372 B	373 B	374 B	375 B	376 B	377 B	378 B	379 B	380 B
381 B	382 B	383 B	384 B	385 B	386 B	387 B	388 B	389 B	390 B	391 B	392 B	393 B	394 B	395 B	396 B	397 B	398 B	399 B	400 B

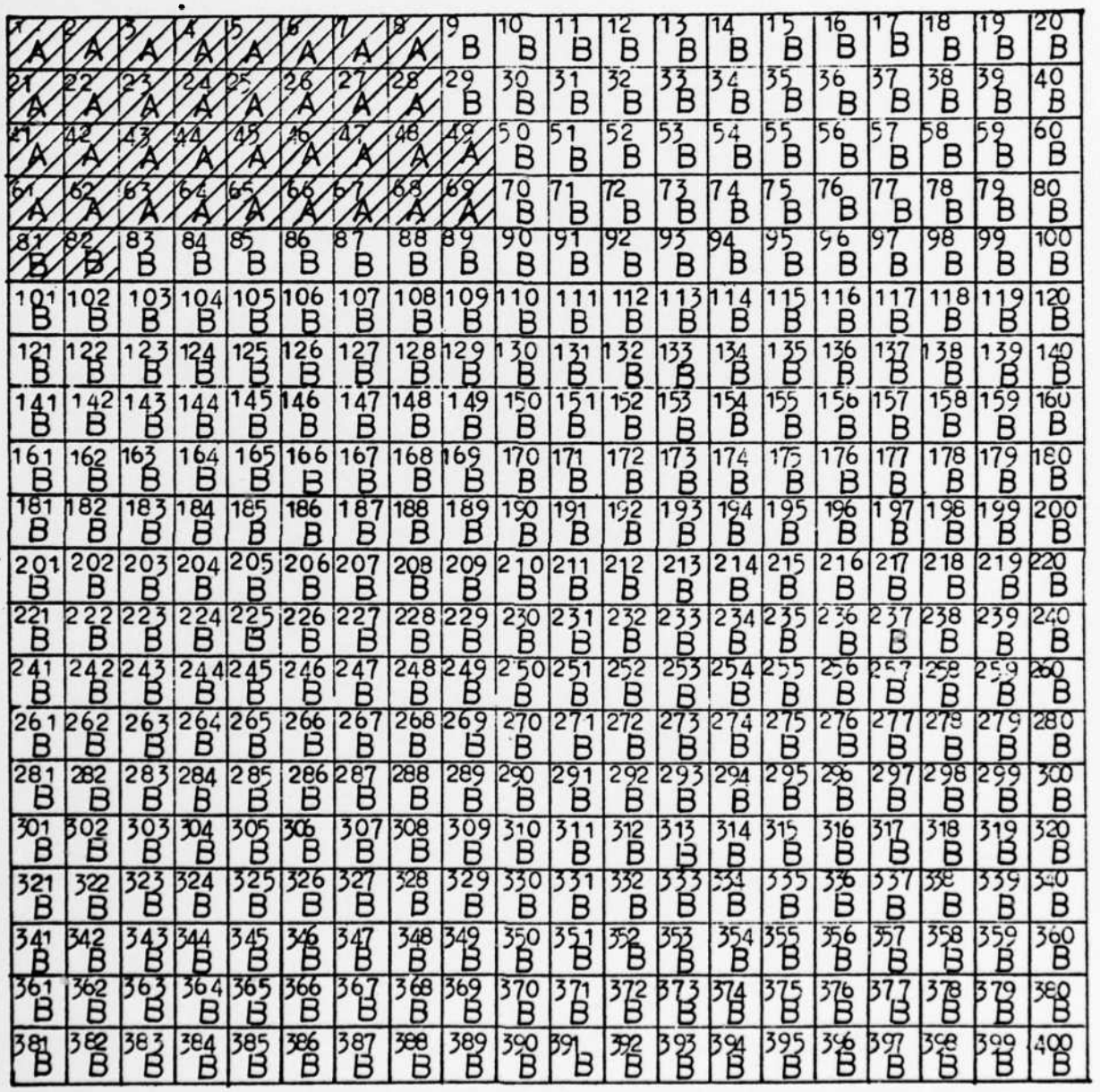
Note: letters represent clusters

Figure 19

Data Set 1

K^{th} Nearest Neighbor

K = 25



Note: letters represent clusters

Figure 20

Data Set 1

K^{th} Nearest Neighbor

K = 30

1 A	2 A	3 A	4 A	5 A	6 A	7 A	8 B	9 B	10 B	11 B	12 B	13 B	14 B	15 B	16 B	17 B	18 B	19 B	20 B
21 A	22 A	23 A	24 A	25 A	26 A	27 A	28 A	29 B	30 B	31 B	32 B	33 B	34 B	35 B	36 B	37 B	38 B	39 B	40 B
41 A	42 A	43 A	44 A	45 A	46 A	47 A	48 A	49 B	50 B	51 B	52 B	53 B	54 B	55 B	56 B	57 B	58 B	59 B	60 B
61 A	62 A	63 A	64 A	65 A	66 A	67 A	68 A	69 A	70 B	71 B	72 B	73 B	74 B	75 B	76 B	77 B	78 B	79 B	80 B
81 B	82 B	83 B	84 B	85 B	86 B	87 B	88 B	89 B	90 B	91 B	92 B	93 B	94 B	95 B	96 B	97 B	98 B	99 B	100 B
101 B	102 B	103 B	104 B	105 B	106 B	107 B	108 B	109 B	110 B	111 B	112 B	113 B	114 B	115 B	116 B	117 B	118 B	119 B	120 B
121 B	122 B	123 B	124 B	125 B	126 B	127 B	128 B	129 B	130 B	131 B	132 B	133 B	134 B	135 B	136 B	137 B	138 B	139 B	140 B
141 B	142 B	143 B	144 B	145 B	146 B	147 B	148 B	149 B	150 B	151 B	152 B	153 B	154 B	155 B	156 B	157 B	158 B	159 B	160 B
161 B	162 B	163 B	164 B	165 B	166 B	167 B	168 B	169 B	170 B	171 B	172 B	173 B	174 B	175 B	176 B	177 B	178 B	179 B	180 B
181 B	182 B	183 B	184 B	185 B	186 B	187 B	188 B	189 B	190 B	191 B	192 B	193 B	194 B	195 B	196 B	197 B	198 B	199 B	200 B
201 B	202 B	203 B	204 B	205 B	206 B	207 B	208 B	209 B	210 B	211 B	212 B	213 B	214 B	215 B	216 B	217 B	218 B	219 B	220 B
221 B	222 B	223 B	224 B	225 B	226 B	227 B	228 B	229 B	230 B	231 B	232 B	233 B	234 B	235 B	236 B	237 B	238 B	239 B	240 B
241 B	242 B	243 B	244 B	245 B	246 B	247 B	248 B	249 B	250 B	251 B	252 B	253 B	254 B	255 B	256 B	257 B	258 B	259 B	260 B
261 B	262 B	263 B	264 B	265 B	266 B	267 B	268 B	269 B	270 B	271 B	272 B	273 B	274 B	275 B	276 B	277 B	278 B	279 B	280 B
281 B	282 B	283 B	284 B	285 B	286 B	287 B	288 B	289 B	290 B	291 B	292 B	293 B	294 B	295 B	296 B	297 B	298 B	299 B	300 B
301 B	302 B	303 B	304 B	305 B	306 B	307 B	308 B	309 B	310 B	311 B	312 B	313 B	314 B	315 B	316 B	317 B	318 B	319 B	320 B
321 B	322 B	323 B	324 B	325 B	326 B	327 B	328 B	329 B	330 B	331 B	332 B	333 B	334 B	335 B	336 B	337 B	338 B	339 B	340 B
341 B	342 B	343 B	344 B	345 B	346 B	347 B	348 B	349 B	350 B	351 B	352 B	353 B	354 B	355 B	356 B	357 B	358 B	359 B	360 B
361 B	362 B	363 B	364 B	365 B	366 B	367 B	368 B	369 B	370 B	371 B	372 B	373 B	374 B	375 B	376 B	377 B	378 B	379 B	380 B
381 B	382 B	383 B	384 B	385 B	386 B	387 B	388 B	389 B	390 B	391 B	392 B	393 B	394 B	395 B	396 B	397 B	398 B	399 B	400 B

Note: letters represent clusters

Figure 21

Data Set 1

K^{th} Nearest Neighbor

K = 40

1	2	3	4	5	6	7	8	9 A	10 A	11 A	12 A	13 A	14 A	15 A	16 A	17 A	18 A	19 A	20 A
21	22	23	24	25	26	27	28	29	30 A	31 A	32 A	33 A	34 A	35 A	36 A	37 A	38 A	39 A	40 A
41	42	43	44	45	46	47	48	49	50 A	51 A	52 A	53 A	54 A	55 A	56 A	57 A	58 A	59 A	60 A
61	62	63	64	65	66	67	68	69	70 A	71 A	72 A	73 A	74 A	75 A	76 A	77 A	78 A	79 A	80 A
81	82	83 A	84 A	85 A	86 A	87 A	88 A	89 A	90 A	91 A	92 A	93 A	94 A	95 A	96 A	97 A	98 A	99 A	100 A
101 A	102 A	103 A	104 A	105 A	106 A	107 A	108 A	109 A	110 A	111 A	112 A	113 A	114 A	115 A	116 A	117 A	118 A	119 A	120 A
121 A	122 A	123 A	124 A	125 A	126 A	127 A	128 A	129 A	130 A	131 A	132 A	133 A	134 A	135 A	136 A	137 A	138 A	139 A	140 A
141 A	142 A	143 A	144 A	145 A	146 A	147 A	148 A	149 A	150 A	151 A	152 A	153 A	154 A	155 A	156 A	157 A	158 A	159 A	160 A
161 A	162 A	163 A	164 A	165 A	166 A	167 A	168 A	169 A	170 A	171 A	172 A	173 A	174 A	175 A	176 A	177 A	178 A	179 A	180 A
181 A	182 A	183 A	184 A	185 A	186 A	187 A	188 A	189 A	190 A	191 A	192 A	193 A	194 A	195 A	196 A	197 A	198 A	199 A	200 A
201 A	202 A	203 A	204 A	205 A	206 A	207 A	208 A	209 A	210 A	211 A	212 A	213 A	214 A	215 A	216 A	217 A	218 A	219 A	220 A
221 A	222 A	223 A	224 A	225 A	226 A	227 A	228 A	229 A	230 A	231 A	232 A	233 A	234 A	235 A	236 A	237 A	238 A	239 A	240 A
241 A	242 A	243 A	244 A	245 A	246 A	247 A	248 A	249 A	250 A	251 A	252 A	253 A	254 A	255 A	256 A	257 A	258 A	259 A	260 A
261 A	262 A	263 A	264 A	265 A	266 A	267 A	268 A	269 A	270 A	271 A	272 A	273 A	274 A	275 A	276 A	277 A	278 A	279 A	280 A
281 A	282 A	283 A	284 A	285 A	286 A	287 A	288 A	289 A	290 A	291 A	292 A	293 A	294 A	295 A	296 A	297 A	298 A	299 A	300 A
301 A	302 A	303 A	304 A	305 A	306 A	307 A	308 A	309 A	310 A	311 A	312 A	313 A	314 A	315 A	316 A	317 A	318 A	319 A	320 A
321 A	322 A	323 A	324 A	325 A	326 A	327 A	328 A	329 A	330 A	331 A	332 A	333 A	334 A	335 A	336 A	337 A	338 A	339 A	340 A
341 A	342 A	343 A	344 A	345 A	346 A	347 A	348 A	349 A	350 A	351 A	352 A	353 A	354 A	355 A	356 A	357 A	358 A	359 A	360 A
361 A	362 A	363 A	364 A	365 A	366 A	367 A	368 A	369 A	370 A	371 A	372 A	373 A	374 A	375 A	376 A	377 A	378 A	379 A	380 A
381 A	382 A	383 A	384 A	385 A	386 A	387 A	388 A	389 A	390 A	391 A	392 A	393 A	394 A	395 A	396 A	397 A	398 A	399 A	400 A

Note: letters represent clusters

well the K^{th} nearest neighbor clustering procedure performs and how various values of K affect the results.

When K = 10 (Figure 16) the procedure produces four clusters. Cluster A appears to identify the lightest pixels in Figure 1 which is the red (hatched) area in Figure 2. Cluster A and the red area match exactly. Cluster B appears from Figure 1 to be the darkest pixels and Clusters C and D appear to fall somewhere in between.

When K = 15 (Figure 17) the procedure produces only two clusters. Cluster A corresponds exactly to the red area in Figure 2. Cluster B is then the darker, non-red area.

When K = 20 (Figure 18) the results are almost exactly the same as for K = 15 except pixel 81 is now in Cluster B. For K = 25 (Figure 19), pixel 82 migrates to Cluster B. For K = 30 (Figure 20), pixels 49 and 8 also migrate to Cluster B.

Finally, for K = 40 (Figure 21) the tree revealed only one high-density cluster. However, many of the pixels in the red area of Figure 2 were "leftover" in the sense that they were included in the one cluster at the very end of the single linkage procedure. In effect, the procedure identified these pixels as outliers of the cluster.

6.2.2 Data Set 2 Results: Figures 22 through 27 show the constrained clusters for Data Set 2 after applying the K^{th} nearest neighbor clustering procedure for selected values of K (K = 10, 15, 20, 25, 30, and 40 respectively).

A comparison similar to that of section 6.2.1 will be made between Figures 1 and 3 and Figures 22 through 27 to qualitatively evaluate the results and differences.

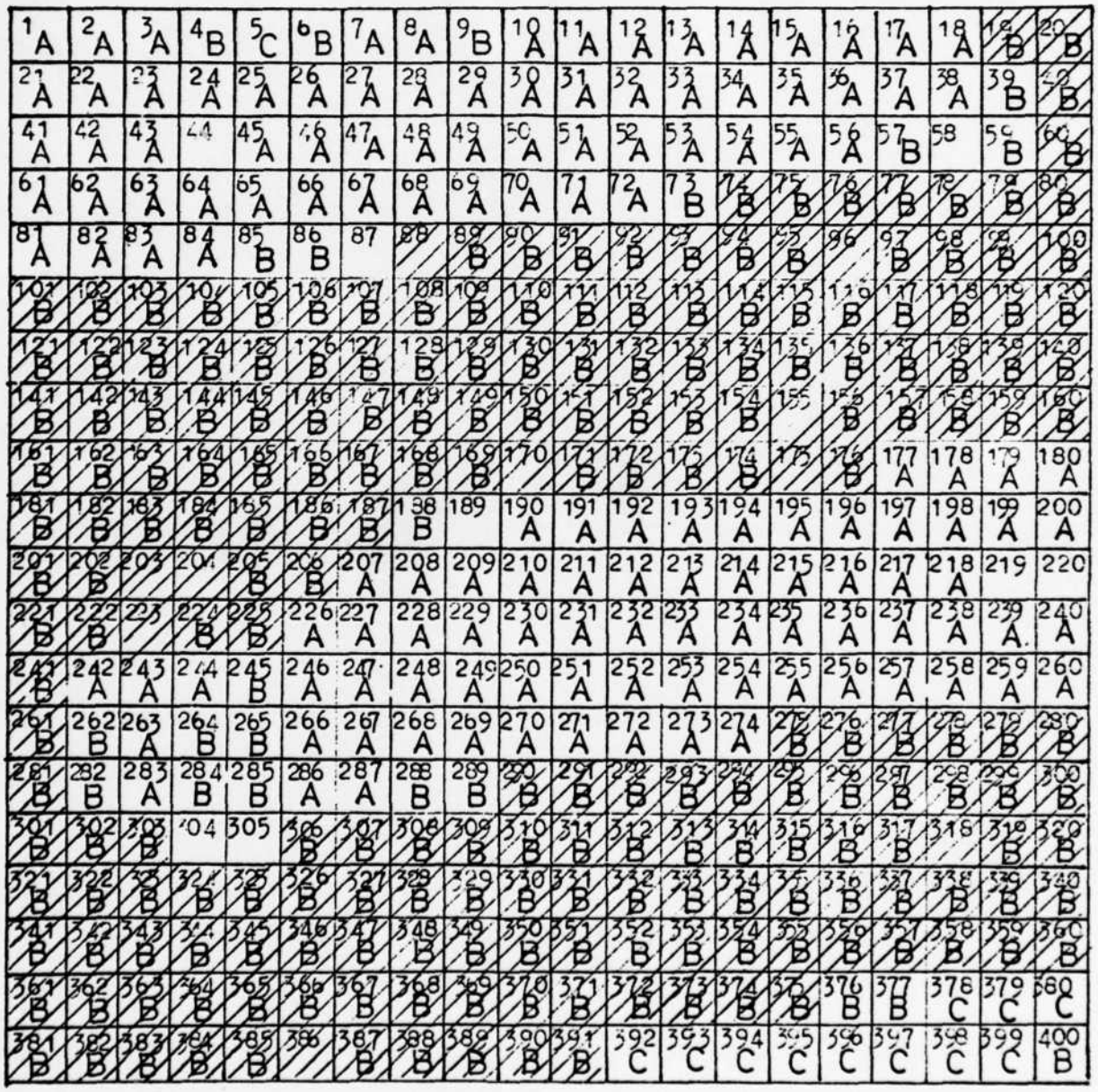
When K = 10 (Figure 22) a comparison with Figure 3 shows three clusters that more or less follow the boundaries of the red (hatched) area in Figure 3. Cluster B appears

Figure 22

Data Set 2

K^{th} Nearest Neighbor

K = 10



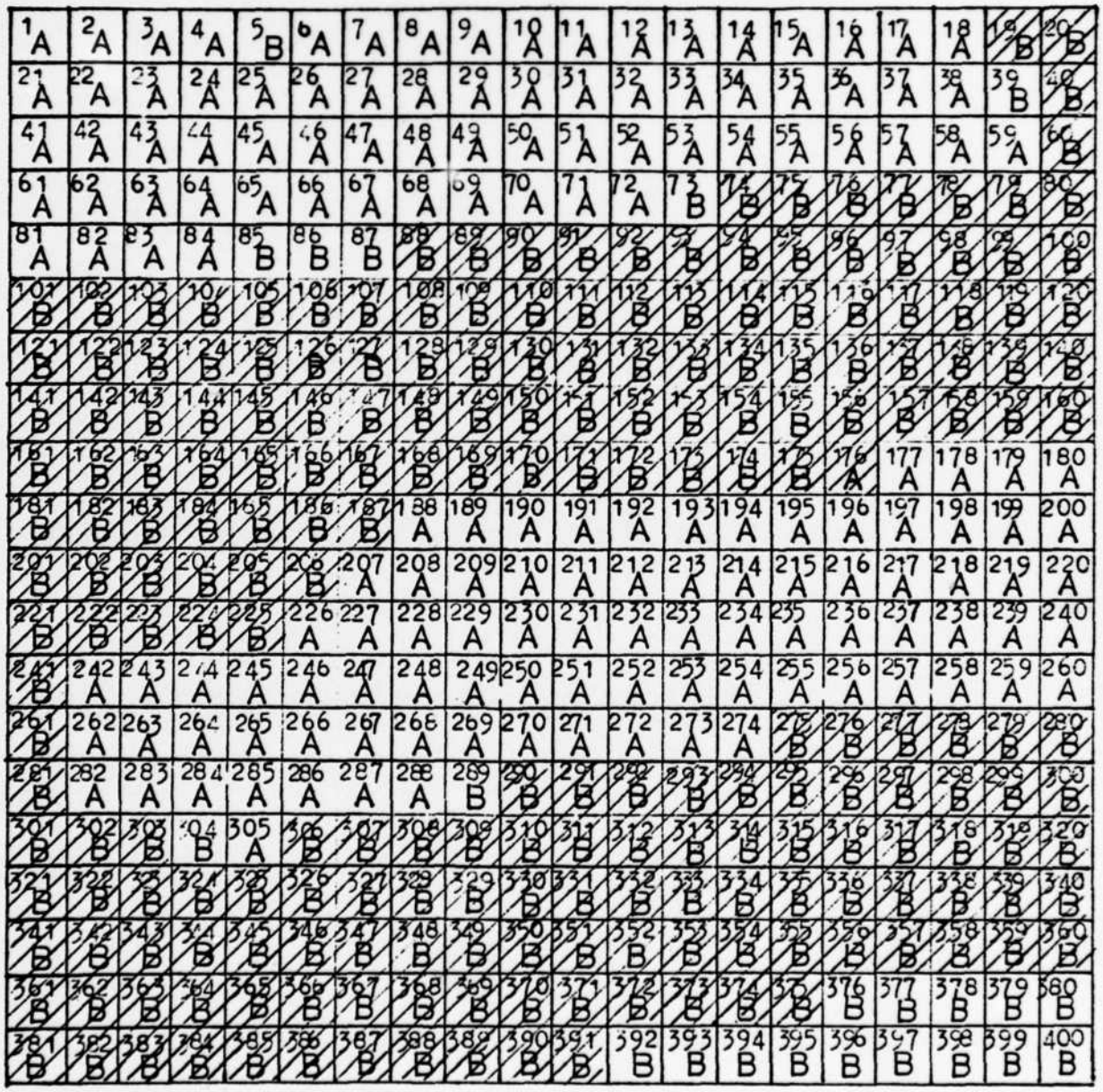
Note: letters represent clusters

Figure 23

Data Set 2

K^{th} Nearest Neighbor

K = 15



Note: letters represent clusters

Figure 24

Data Set 2

K^{th} Nearest Neighbor

K = 20

1 A	2 A	3 A	4	5	6	7 A	8 A	9 A	10 A	11 A	12 A	13 A	14 A	15 A	16 A	17 A	18 A	19 B	20 B
21 A	22 A	23 A	24 A	25 A	26 A	27 A	28 A	29 A	30 A	31 A	32 A	33 A	34 A	35 A	36 A	37 A	38 A	39	40 B
41 A	42 A	43 A	44	45 A	46 A	47 A	48 A	49 A	50 A	51 A	52 A	53 A	54 A	55 A	56 A	57	58 A	59 A	60 B
61 A	62 A	63 A	64 A	65 A	66 A	67 A	68 A	69 A	70 A	71 A	72 A	73 B	74 B	75 B	76 B	77 B	78 B	79 B	80 B
81 A	82 A	83 A	84 A	85	86	87	88	89 B	90 B	91 B	92 B	93 B	94	95 B	96	97 B	98 B	99 B	100 B
101 B	102 B	103 B	104 B	105 B	106 B	107 B	108 B	109 B	110 B	111 B	112 B	113	114	115	116	117	118	119	120
121 B	122 B	123 B	124 B	125 B	126 B	127 B	128 B	129 B	130 B	131	132	133	134	135 B	136	137	138	139 B	140 B
141 B	142 B	143 B	144 B	145 B	146 B	147 B	148 B	149 B	150	151	152 B	153	154 B	155 B	156 B	157	158 B	159 B	160 B
161 B	162 B	163 B	164	165	166 B	167 B	168 B	169	170	171 B	172 B	173 B	174 B	175	176 A	177 A	178 A	179 A	180 A
181 B	182 B	183 B	184 B	185 B	186	187	188	189 A	190 A	191 A	192 A	193 A	194 A	195 A	196 A	197 A	198 A	199 A	200 A
201	202	203	204 B	205 B	206	207 A	208 A	209 A	210 A	211 A	212 A	213 A	214 A	215 A	216 A	217 A	218 A	219 A	220 A
221 B	222 B	223	224 B	225	226 A	227 A	228 A	229 A	230 A	231 A	232 A	233 A	234 A	235 A	236 A	237 A	238 A	239 A	240 A
241 B	242 A	243 A	244 A	245	246 A	247 A	248 A	249 A	250 A	251 A	252 A	253 A	254 A	255 A	256 A	257 A	258 A	259 A	260 A
261 B	262	263 A	264	265	266 A	267 A	268 A	269 A	270 A	271 A	272 A	273 A	274 A	275	276	277	278 B	279 B	280 B
281 B	282	283 A	284	285	286 A	287 A	288	289	290	291	292	293 B	294 B	295 B	296 B	297 B	298 B	299 B	300 B
301 B	302 B	303 B	304	305	306 B	307 B	308 B	309 B	310 B	311 B	312 B	313 B	314 B	315 B	316 B	317 B	318 B	319	320 B
321 B	322 B	323 B	324 B	325	326 B	327 B	328 B	329 B	330 B	331 B	332 B	333 B	334 B	335 B	336 B	337 B	338 B	339 B	340
341 B	342 B	343 B	344 B	345 B	346 B	347 B	348 B	349	350 B	351 B	352	353 B	354 B	355 B	356 B	357 B	358 B	359 B	360 B
361 B	362 B	363 B	364 B	365 B	366 B	367 B	368 B	369	370 B	371 B	372 B	373 B	374 B	375 B	376	377	378	379	380
381 B	382 B	383 B	384 B	385	386 B	387 B	388 B	389 B	390 B	391 B	392	393	394	395	396	397	398	399	400

Note: letters represent clusters

Figure 25

Data Set 2

K^{th} Nearest Neighbor

K = 25

1 A	2 A	3 A	4 A	5 B	6 A	7 A	8 A	9 A	10 A	11 A	12 A	13 A	14 A	15 A	16 A	17 A	18 A	19 B	20 B
21 A	22 A	23 A	24 A	25 A	26 A	27 A	28 A	29 A	30 A	31 A	32 A	33 A	34 A	35 A	36 A	37 A	38 A	39 A	40 B
41 A	42 A	43 A	44 A	45 A	46 A	47 A	48 A	49 A	50 A	51 A	52 A	53 A	54 A	55 A	56 A	57 A	58 A	59 A	60 B
61 A	62 A	63 A	64 A	65 A	66 A	67 A	68 A	69 A	70 A	71 A	72 A	73 B	74 B	75 B	76 B	77 B	78 B	79 B	80 B
81 A	82 A	83 A	84 A	85 A	86 A	87 B	88 B	89 B	90 B	91 B	92 B	93 B	94 B	95 B	96 B	97 B	98 B	99 B	100 B
101 B	102 B	103 B	104 B	105 B	106 B	107 B	108 B	109 B	110 B	111 B	112 B	113 B	114 B	115 B	116 B	117 B	118 B	119 B	120 B
121 B	122 B	123 B	124 B	125 B	126 B	127 B	128 B	129 B	130 B	131 B	132 B	133 B	134 B	135 B	136 B	137 B	138 B	139 B	140 B
141 B	142 B	143 B	144 B	145 B	146 B	147 B	148 B	149 B	150 B	151 B	152 B	153 B	154 B	155 B	156 B	157 B	158 B	159 B	160 B
161 B	162 B	163 B	164 B	165 B	166 B	167 B	168 B	169 B	170 B	171 B	172 B	173 B	174 B	175 B	176 A	177 A	178 A	179 A	180 A
181 B	182 B	183 B	184 B	185 B	186 B	187 B	188 A	189 A	190 A	191 A	192 A	193 A	194 A	195 A	196 A	197 A	198 A	199 A	200 A
201 B	202 B	203 B	204 B	205 B	206 B	207 A	208 A	209 A	210 A	211 A	212 A	213 A	214 A	215 A	216 A	217 A	218 A	219 A	220 A
221 B	222 B	223 B	224 B	225 B	226 A	227 A	228 A	229 A	230 A	231 A	232 A	233 A	234 A	235 A	236 A	237 A	238 A	239 A	240 A
241 B	242 A	243 A	244 A	245 A	246 A	247 A	248 A	249 A	250 A	251 A	252 A	253 A	254 A	255 A	256 A	257 A	258 A	259 A	260 A
261 B	262 A	263 A	264 A	265 A	266 A	267 A	268 A	269 A	270 A	271 A	272 A	273 A	274 A	275 A	276 B	277 B	278 B	279 B	280 B
281 B	282 A	283 A	284 A	285 A	286 A	287 A	288 A	289 A	290 A	291 B	292 B	293 B	294 B	295 B	296 B	297 B	298 B	299 B	300 B
301 B	302 B	303 B	304 A	305 A	306 B	307 B	308 B	309 B	310 B	311 B	312 B	313 B	314 B	315 B	316 B	317 B	318 B	319 B	320 B
321 B	322 B	323 B	324 B	325 B	326 B	327 B	328 B	329 B	330 B	331 B	332 B	333 B	334 B	335 B	336 B	337 B	338 B	339 B	340 B
341 B	342 B	343 B	344 B	345 B	346 B	347 B	348 B	349 B	350 B	351 B	352 B	353 B	354 B	355 B	356 B	357 B	358 B	359 B	360 B
361 B	362 B	363 B	364 B	365 B	366 B	367 B	368 B	369 B	370 B	371 B	372 B	373 B	374 B	375 B	376 B	377 B	378 B	379 B	380 B
381 B	382 B	383 B	384 B	385 B	386 B	387 B	388 B	389 B	390 B	391 B	392 B	393 B	394 B	395 B	396 B	397 B	398 B	399 B	400 B

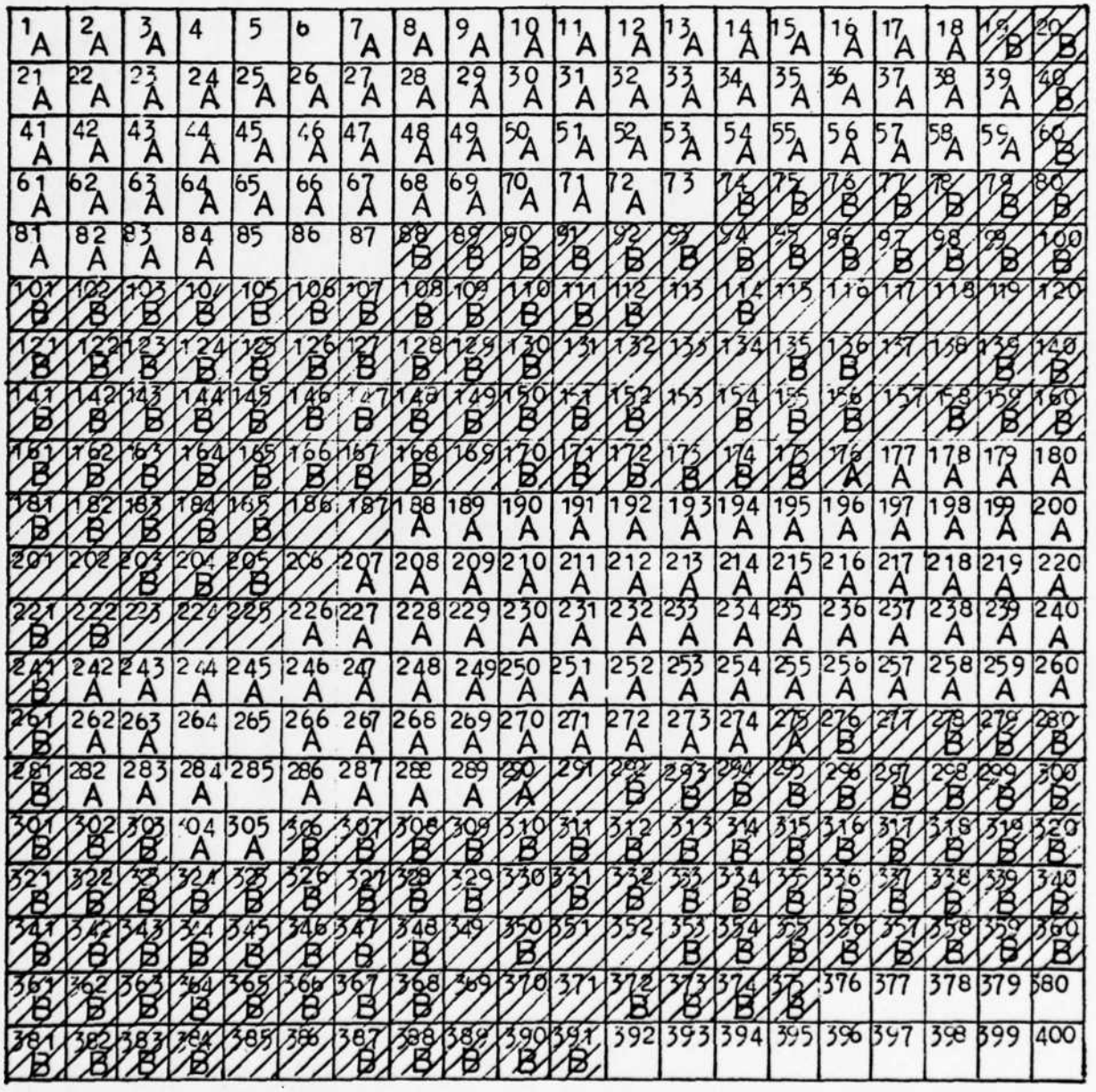
Note: letters represent clusters

Figure 26

Data Set 2

K^{th} Nearest Neighbor

K = 30



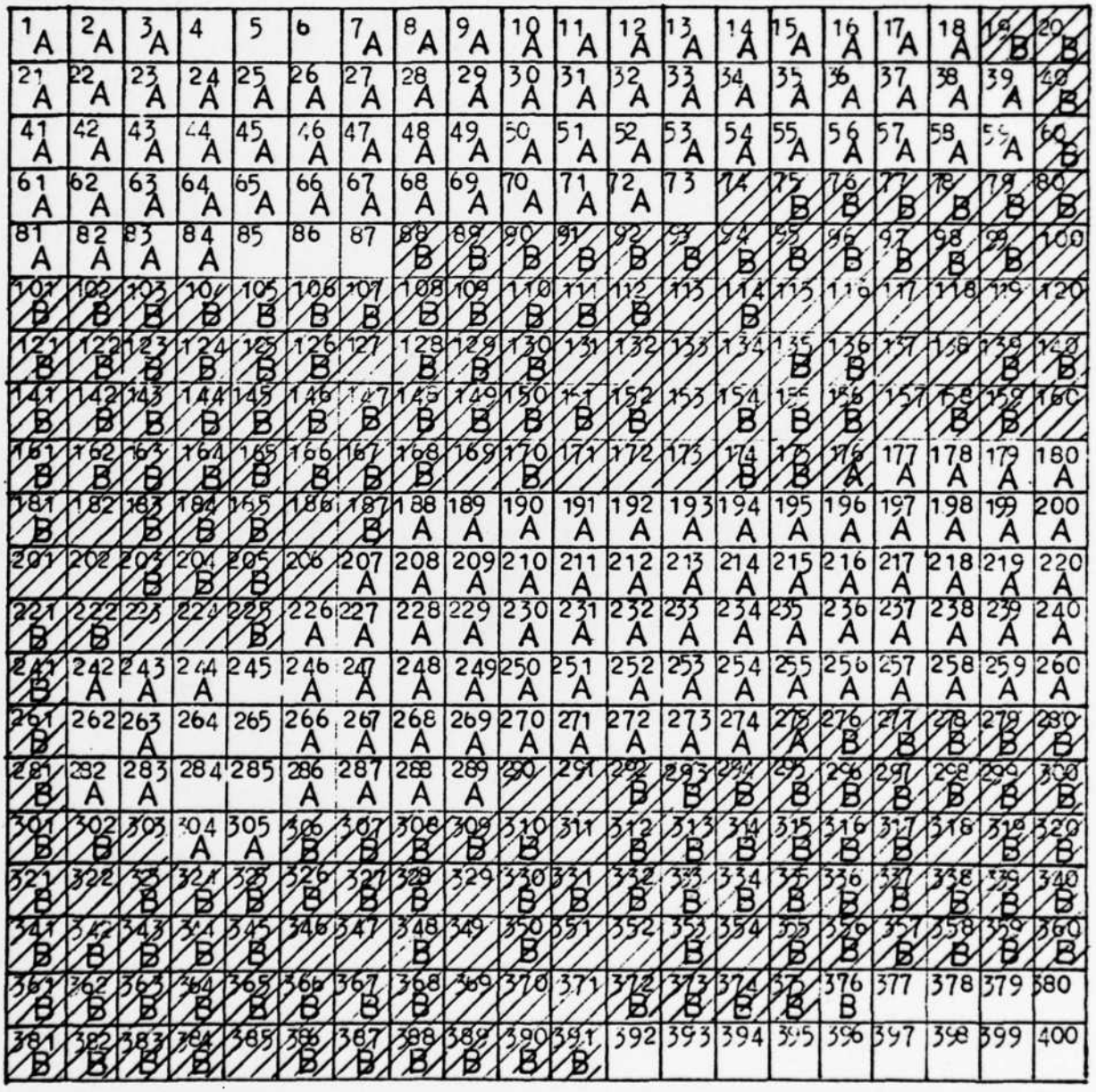
Note: letters represent clusters

Figure 27

Data Set

K^{th} Nearest Neighbor

K = 40



Note: letters represent clusters

to be the red area, except pixels 4, 6, 9, 39, 57, 59, 73, 85, 86, 188, 245, 262, 264, 265, 282, 284, 285, 288 and 289 spill over into the non-red area. Cluster C appears to be the light non-red area, and Cluster A appears to be the dark (green) area. Pixels 376, 377, and 400 are in Cluster B instead of Cluster C. In this case, there are quite a few "leftover" pixels (e.g., pixels 87, 88, 304 and 305). Here, "leftover" means that these pixels were not included in any one of the clusters during the single linkage cluster formations, but were included in the cluster tree only after two or more of the already formed clusters were combined into a larger single cluster. For example, at some density value f_0, Clusters A and B might be combined to form one large cluster incorporating all their pixels and some of the "leftover" pizels.

When K = 15 (Figure 23), the procedure produces only two clusters, Clusters A and B. Cluster B has "swallowed up" Cluster C of Figure 22 and represents all the lighter pizels as seen in Figure 2. Here, pixels 39, 73, 85, 86, 87, 289, and 304 have spilled over into Cluster B and 176 has spilled over into Cluster A when compared with Figures 1 and 3. Of these "misplaced" pixels, only pixel 176 changed clusters from Figure 22 when K = 10. The rest were previously identified as "misplaced" or "leftover" in Figure 22. Since there are only two clusters, red and light (white or grey) are not distinguishable for K = 15 using this procedure.

For K = 20 (Figure 24) and for all larger K, there are only two clusters produced by the procedure. In comparing Figure 24 with Figure 3, only pixel 73 has spilled over into Cluster B and only pixel 176 into Cluster A. Both were previously identified. However, there are many more "leftover" pixels.

For K = 25 (Figure 25), the "leftover" pixels were assigned to either cluster based on which cluster they were closest to in the tree of clusters, i.e. the smaller distance in the distance matrix from them to the two clusters during the single linkage amalgamation process. In this case pixels 73 and 87 spill over into Cluster B, and 176 and 290 spill over into A. Both 87 and 290 were "leftover" pixels for K = 20.

The only differences between K = 30 (Figure 26) and K = 25 (Figure 25) are that pixels 73 and 87 become "leftover", and the only difference between K = 30 (figure 26) and K = 40 (Figure 27) is pixel 290 becomes "leftover". In addition, it is easy to see there are more "leftover" modes as K increases.

6.2.3 Diagnostic Plot Analysis: Figure 28 provides the diagnostic plot for Data Set 1. The "elbow" appears to occur at K = 15 with two modes and is stable until K = 40. Based on the previous discussion of these plots, there appears to be two modes (clusters) in this population (Data Set).

Figure 29 provides the diagnostic plot for Data Set 2. There are "elbows" at both K = 6 with 3 modes and K = 15 with two modes. Figure 1 appears to support 2 modes but the original color photograph supported 3 modes, i.e., red, green, and white or grey.

Then using the number of modes to represent the number of clusters, Figures 28 and 29 show that there are two clusters in Data Set 1 and two or three clusters in Data Set 2. From Figure 1, it appears there are two or three shades in Data Set 1 and three in Data Set 2. Therefore, the diagnostic plot appears to have been accurate in these cases.

FIGURE 28
DIAGNOSTIC PLOT
DATA SET 1

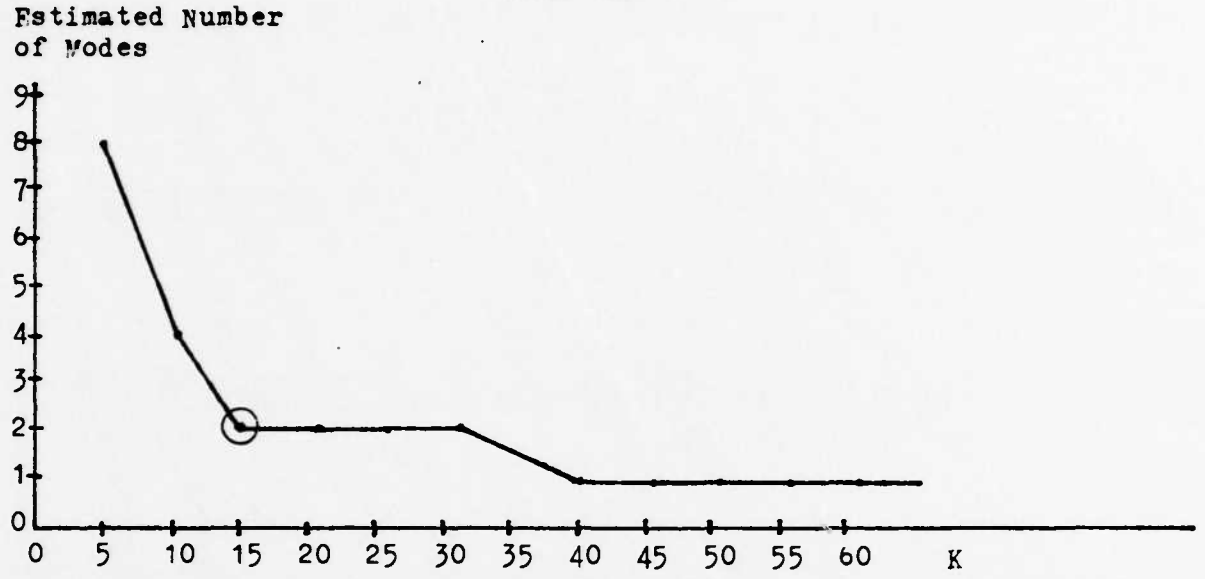
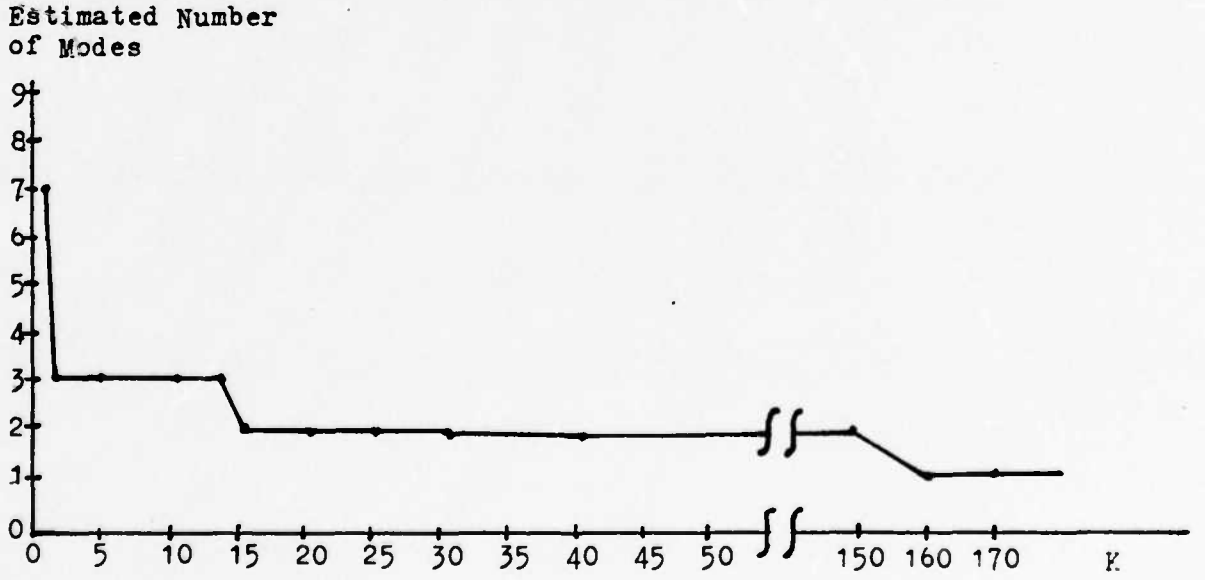


FIGURE 29
DIAGNOSTIC PLOT
DATA SET 2



6.3 Method Evaluation and Problems: It was observed in Fiugres 16 to 27 that some of the pixels were amalgamated into clusters they should not have been according to Figures 2 and 3. However, it must be remembered that Figures 2 and 3 were constructed "by eye" and decisions as to which pixels were red and non-red were somewhat subjective. In addition, all of the pixels observed as spilling over into other incorrect clusters were either border pixels or sometimes categorized as "leftover". Border pixels are on the red, non-red area borders. The categorization of these types of pixels in Figures 2 and 3 was the most difficult. On the other hand, "leftover" pixels were those that did not clearly belong to any one cluster in the first place. Therefore, all things considered, the K^{th} nearest neighbor clustering procedure very accurately assigned pixels to their appropriate clusters.

As stated previously, the diagnostic plots of the estimated number of modes versus K were useful in determining the number of sample population modes. They appeared accurate in suggesting the number of modes in each case. However, as seen with Data Set 2, these plots can lead to ambiguities as to what the actual number of modes should be (two or three for Data Set 2). This is a weakness of the diagnostic plot, and further work needs to be done to overcome such situations, or at least provide guidance in making final decisions (e.g. further tests).

Another problem with this procedure remains with the final choice of K. As seen, for ranges of K over which the estimated number of modes remains the same, different K result in different cluster assignments for some pixels. The membership of pixels into the high-density clusters appears dependent on the value of K chosen. Perhaps results must be examined over this entire range of K before final selections can be made.

Compared to the K-Means method, this method has a disadvantage of "leftover" pixels. However, this is not a major problem or weakness of the system, because there are many ways in which to assign such pixels to the clusters formed (one example was given in section 6.2.2).

A final weakness or limitation of this procedure is that it does not use all the available information. That is, this method, as with the K-Means method, does not take geographical information into account. Distances between pixels and adjacency of pixels is not considered when forming clusters. Instead, clusters are formed strictly on their multi-spectral measurements. Then, the "geographically-constrained" clusters are based on contiguous batches of pixels within the same spectral cluster. The next method, studied in Chapter 7, attempts to overcome this void by including geographical information in its clustering algorithm.

Chapter 7

The Graph Decomposition Method

7.0 Introduction: This chapter describes a graph decomposition technique for solving the constrained-clustering problem. The graph decomposition technique employs a high-density clustering model on graphs (see Wong and Latten, 1982). Chapter content consists of an introduction to this method, the results of applying the method to the two data sets, and a discussion of the method's strengths and weaknesses.

The introductory material for this method includes its usefulness in solving general clustering problems; some preliminary definitions, a survey of existing clustering techniques, a description of the high-density clustering model, and its accompanying algorithm. It should be noted that this method and its application to the constrained clustering problem are the heart of this thesis. The two previous methods serve as a baseline and frame of reference for evaluating the graph decomposition technique.

7.1 The Method's Application to a General Class of Problems: The graph decomposition method is applicable to a wide vareity of applications including not only constrained clustering problems, but also more general situations involving large systems consisting of many elements which are interrelated. In these situations, it is often desirable and useful to break up these systems into smaller, more manageable subsystems.

Complex systems, be they physical, data, or more general systems, are characterized by complex integrations and interactions between competing and complementary subsystems. A detailed analysis of such a system as a whole

is often beyond the scope of an analyst's conceptual abilities. Analysts often attempt to cope with this difficulty by decomposing the original system into smaller, more conceptually manageable subsystems or subproblems. Often, the analyst's decomposition approach is largely intuitive and subjective.

In presenting the graph decomposition technique, a systematic approach is proposed for decomposing an overall system into more coherent, manageable subsystems with a structure exhibiting strong coupling within subsystems and weak coupling between them.

In order to implement this approach, the analyst must complete a preliminary task. The task involves identifying the basic elements of the system and examining each pair of elements to decide whether a significant degree of interdependence exists between them. In other words, the analyst must decide if interaction (either interference or support) exists between each pair. Then, he must assign weights to each relationship based on the degree of interaction he perceives. The weights could be as simple as interdependence versus no interdependence (weights of 1 or 0), or they could be more detailed, subjective weights based on degree of dependence ranging from complete independence (0,0) to full interdependence (1,0), for example. While this step is subjective and involves considerable intuition and judgement, it is unavoidable in any nontrivial analysis. How to perform such a task is not discussed here and will be assumed completed in what follows.

Once the basic elements of the system have been identified and weights have been assigned to interdependencies between element pairs, the entire system can be represented as an undirected graph (defined in section 7.2.1). Such a

graph consists of numbered nodes (the basic elements of the system) connected by links (the weights assigned to pairs of elements). The decomposition task is then reduced to finding subgraphs or clusters of nodes (subsystems) which are densely connected internally and separated from each other by relatively few or weak cross-links.

The proposed implementation of the decomposition task is via a high-density clustering method for graphs. The model and algorithms for this method are described in Sections 7.4 and 7.5 with test examples described in Appendix A. Preliminary definitions and discussion with a short review of the existing theory are given in Section 7.2, followed by a brief and related description of some existing clustering techniques in Section 7.3.

This graph partitioning method, based on the high-density clustering model for graphs was developed and is described in detail by Wong and Lattin (1982). Their paper also includes the environment surrounding and motivation for developing the technique and references for further information, and many of the preliminaries in this chapter are extracted from it.

7.2 Preliminary Definitions and Discussions:

7.2.1 Definitions:

Some of the definitions given here are not general in that they are tailored for the discussion to follow.

a) Graph: A collection of objects called nodes which are interconnected by a network of paths. Each segment of such a path connecting a pair of nodes by only one edge (line segment) is called a link. A graph may be weighted or unweighted and directed or undirected as described below. It is called connected if there exists a path between any pair of nodes.

b) Link: The device (a line segment) used to connect a pair of nodes. Links are used to describe relations between node pairs. These relationships may be interdependencies, distances (not necessarily euclidean) between nodes, similarities, or more general relationships.

c) Weighted graph: A graph is called weighted if the individual links connecting its nodes are assigned values, between 0 and 1 inclusive, for example. A graph whose links are always assigned the values 0 or 1 will be called unweighted. A pair of nodes connected by a link with value 0 can be considered unconnected, and the link will be omitted from the graph.

d) Directed graph: A graph whose links include an implied direction as in a one-way street. Otherwise a graph is called undirected. The graph decomposition technique is mainly concerned with undirected graphs.

e) Tree: A tree is a connected, undirected graph with no cycles, i.e., no circuits where one could return to a previously visited node traversing no link twice. Equivalently, any undirected graph is a tree if:

(i) every pair of nodes is connected by one and only one path.

(ii) all nodes are connected via some path, but if one link is deleted, the resulting graph is no longer connected, i.e., is broken into two separate graphs.

(iii) the graph has no cycles, and if a link is added to join two nodes, one and only one cycle is formed.

f) Minimum spanning tree: A minimum spanning tree of a connected, undirected graph is a collection of links providing a unique path between any pair of nodes such that

the sum of the values assigned to the links is minimum over all trees which contain all the nodes.

g) Cluster: A collection of nodes grouped into a single identifiable unit (subgraph, subtree, subsystem, subproblem, subarea, etc...).

h) Tree of clusters: A tree of clusters is a family T of clusters including the complete set of nodes such that if C and D are elements of T, then the intersection of C and D is either C, D, or the null set (ϕ). (C and D are sets of nodes.)

i) Density contour cluster: Given a density function f, a density contour cluster at level f* is a maximal, connected set of the form $\{x|f(x) \geq f^*\}$.

j) Ultrametric: A distance or similarity measure $P(\cdot,\cdot)$ is called an ultrametric if for any three points x, y, z: $P(x,y) \leq \max [P(x,z), P(y,z)]$.

k) Density contour on a link: The density contour on the link between two linked nodes i and j is defined for an unweighted graph as $dc(i,j) = \frac{\text{Intersection } N(i,j)}{\text{Union } N(i,j)} =$

$$\frac{\text{number of nodes connected to both i and j (i and j included)}}{\text{number of nodes connected to either i and j or both (i and j included)}}$$

and for weighted links as $dc(i,j) = \frac{2W(i,j) + \frac{1}{2}\sum_{k\in C}(W(i,k)+W(j,k))}{\text{Union } N(i,j)}$

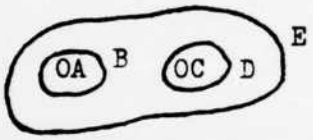
where W(i,j) = the weight on the link between nodes i and j, and C is the set of nodes connected to both i and j, i and j excluded. The contour between two unlinked nodes is defined to be 0.

7.2.2 Preliminary Discussion:

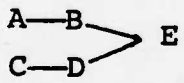
From definitions h) and j), it can easily be shown that an ultrametric determines a tree of clusters of the form $\{z|P(x,z) \leq P(x,y)\}$. Any two clusters of this form are either disjoint or one includes the other.

Conversely, any tree generates an ultrametric by defining $P(x,y)$ as the "level" of the minimal cluster including x and y. Where here "level" could be the density contour of the cluster, the actual level (height) of the subtree containing x and y, etc... Then, an ultrametric can be considered a tree model, and any tree induces an ultrametric which can be extracted from the tree.

From definitions h) and i) above, a density contour diagram such as



contains density contour clusters A,B,C,D, and E and defines a tree of clusters



the above observations will be helpful in understanding the discussion of the high-density cluster model in Section 7.4 and serve as motivation for the model's usefulness.

7.3 Existing Clustering Techniques (A Literature Survey):

7.3.1 Summary of Categories:

This section includes a brief summary of some of the existing clustering categories and techniques, including those more popularly used in forming clusters of nodes (subgraphs) for connected, undirected graphs.

General clustering techniques usually fall under the categories hierarchical, partitioning, density estimating, and block clustering techniques.

Block clustering techniques are not applicable to the constrained clustering problem and will not be discussed further.

Of the partitioning techniques, the "k-means algorithm" (see chapter 5) is the most popular and well known. As explained in Chapter 5, a set of N observations $X_1, X_2 \ldots X_N$ are to be partitioned into K clusters. From a distance point of view, distances within clusters should be small and distances between clusters large. The majority of techniques falling in this category attempt to partition the observations so as to optimize some criterion, such as minimizing the pooled within clusters sum of squares. Although many of the ideas here are applicable to the constrained clustering problem, the solution should not be constrained by some fixed number of clusters. Therefore, a different technique without this deficiency is desirable.

Hierarchical methods, variously referred to as joining or divisive methods include single linkage (nearest neighbor), complete linkage (farthest neighbor), centroid (distances between group centers -- similar to averages) and Ward's method (error sum of squares or ANOVA method).

Each of these methods appears to be effective in certain cases and for particular applications, and not in others. Various arguments have been made in favor of each. For example, some prefer single linkage because it has a continuous ultrametric (i.e., for general distances d, a small change in d changes clusters only slightly), whereas the other methods are discontinuous. Others report complete linkage as more effective in recovering an ultrametric contaminated with errors. Still others recommend average linkage methods based on Monte Carlo studies in which effectiveness was measured by the correlation between true and fitted ultrametrics. None of these methods are optimal under all circumstances.

Density Methods include single linkage, mode analysis, and hybrid algorithms (see Wong, 1982). Of these, single linkage has some interesting properties, some of which are described below in section 7.3, since the graph decomposition method of his chapter and the K^{th} nearest neighbor method of Chapter 6 imply its algorithm. (For a more complete discussion of existing methods, including a literature survey, see Wong and Lattin, 1982.)

7.3.2 Graph Decomposition Techniques:

Subsets and modifications of the above methods have been developed and used for the decomposition of connected graphs. A survey of some of the more successful and well known of these methods is provided in Wong and Lattin (1982). Their conclusion is that none of the existing techniques are particularly well suited for the constrained clustering problem. Some of their discussion is included here to facilitate an understanding of why the graph decomposition method of this chapter was chosen.

Graph decomposition techniques can be broken down into two general types:

(i) Criterion-optimizing techniques where the clusters are found by optimization of some goodness-of-partition criterion, and

(ii) Density-bond-energy, or node-tearing techniques where the clusters are formed by finding strongly connected subgraphs separated by relatively few or relatively weak links.

The majority of type (i) techniques attempt to partition the graph to optimize a predetermined criterion such as minimizing the sum of weights of links between subgraphs. Most of these methods assume a fixed number of existing clusters (a constraint not desirable here) although some

allow the number to be changed during the analysis. With only one exception, these methods do not address the problem of how to determine the proper number of clusters. The one exception does provide a measure of goodness-of-partition (not dicussed here -- see Wong and Lattin, 1982), but this measure favors a large number of subgraphs. Otherwise, these type (i) methods differ mainly on the optimality of their final solution and the efficiency of the employed computational algorithm. They are not considered optimal for the constrained clustering problem because they require the number of clusters to be chosen beforehand.

Type (ii) techniques employ the natural concept of graph clustering, suggesting that parts of the graph (subgraphs) should include densely connected nodes separated by relatively few or relatively weak cross links. This concept will be used to define "natural" or "high density" clusters in Section 7.4. For these methods, each modal subgraph represents a different cluster.

One of the bond-energy algorithms permutes the rows and columns of the graph adjacency matrix (an n by n matrix of the values assigned to the links between the nodes) to push the larger array elements together. The algorithm helps identify the "natural" or "high-density" clusters in a graph as well as associations of these cluster groups with one another (a block clustering technique). Its main drawback is that the best partition is not explicitly given and personal judgment is required to identify the subgraphs.

Node-tearing techniques are best described as attempts to locate sets of nodes (smallest in number) which, when removed, split the graph into two or more unconnected subgraphs. These techniques, in effect, search for nodes not densely connected with other nodes and remove them to reveal the remaining densely connected subgraphs. Their

purpose is to locate "leftover" nodes (low density nodes) rather than locating densely connected modal subgraphs.

Another algorithm identifies leader subgraphs (subgraphs that don't overlap and are strongly coherent). Leftover nodes, if any, are then assigned to these leaders. However, there are problems with assignment rules for left-over nodes, resulting in an incomplete solution. Other drawbacks to techniques in this category include locally optimal solutions (not global solutions) for some, while others can only be applied to unweighted graphs.

It is intuitively clear that density techniques come the closest to addressing the constrained clustering problem. However, none of the existing techniques appears completely satisfactory, and a hybrid of several of these techniques modified to meet the needs of this problem is presented in Section 7.4.

7.3.3 The Single Linkage Method:

As mentioned previously, the single linkage method has several interesting properties that are included here as justification for its future use. (This method is sometimes referred to as the minimum distance, nearest neighbor, and minimum spanning tree method.) The single linkage method can be described and applied as a joining, dividing, partitioning or density method, i.e., it is flexible and provides several ways of looking at the resulting clusters. Its joining algorithm can be characterized as:

(1) Find the "closest" pair of objects or clusters (say i and j)

(2) Join i and j to form a new cluster (i union j)

(3) Eliminate i and j from further consideration as separable entities (elements)

(4) Determine the distance d(i union j,k) between i union j and all other remaining objects and clusters k.

The distance between clusters C and D for this method is defined as $\min_{i,j} \{d(i,j) | i \in C, j \in D\}$

(5) If more than one cluster remains, return to step (1).

Properties of the single linkage method include:

(i) Its associated ultrametric is unique among joining techniques in that it is a continuous function of the given distances (i.e., small changes in distances result in small changes in the clusters or the ultrametric). Other joining techniques have discontinuous ultrametrics with small changes in distances often resulting in large changes in the cluster tree.

(ii) Its clusters are related to nearest neighbor discriminant analysis in which a new object is classified as a member of one of the known classes (clusters) by allocating it to that class containing the closest member to it.

(iii) For a given distance measure d(i,j), fixed value d, and the graph on all objects in which x and y are linked when $d(x,y) \leq d$, the maximally connected subgraphs are the single linkage clusters.

(iv) Its clusters are closely related to the minimum spanning tree (MST) previously defined as the shortest length set of paths covering all nodes. Its algorithm is listed below. Let S be a set of objects and T be a tree.

(1) Begin with any object in S since the formed clusters do not depend on the starting point. Let the object be in the tree.

(2) Find the object in S-T which is closest to an object in T.

(3) Add the object to T. Let N(K) identify the k^{th} object added and let d(k) be its distance to the closest object in T.

(4) Repeat Step (2) until $T = S$

(5) Form single linkage clusters

$N(k_1)$, $N(k_1+1)$,... $N(k_2)$ whenever $\max_{k_1<i\leq k_2} d(i) < d(k_1), d(k_2+1)$.

This algorithm is included here because a modification of it is implemented in the program for finding high-density clusters. (The program, written in Mortran (structured Fortran), is listed in Appendix B.)

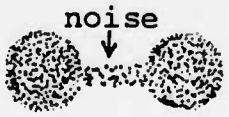
(v) When considered as a density method in one dimension, it asymptotically divides clusters at the minimum of the density and isolates nodes in different single linkage clusters. In higher dimensions, the generalization is not straightforward.

(vi) The method requires a "distance" matrix defining the value of the links. All the information in the distance matrix needed to form the single linkage clusters (the single linkage tree) is contained in the MST of the graph.

(vii) It has a relatively efficient computational algorithm bounded by $0(N^2)$ using the MST technique (see Wong and Lattin, 1982).

(viii) It tends not to form arbitrary clusters as do the other joining methods.

(ix) It has a serious deficiency in that noise can limit its effectiveness. For example, a set of observations of the form



with two coherent, dense clusters connected by noise will simply be "chained" by this method into a single cluster. Modifying this procedure for graphs to include density considerations eliminates this deficiency.

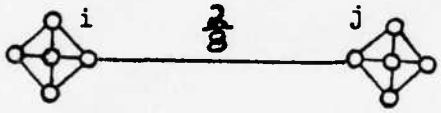
With these single linkage properties as justification (this list is certainly not complete), the high-density

clustering model will now be described.

7.4 The High-Density Clustering Model:

For a given graph, clusters of nodes may be thought of as densely-connected subgraphs separated from other subgraphs by relatively few links.

Intuitively, two nodes should not belong to the same cluster simply because they are linked as shown below.



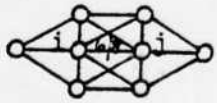
(i and j should not be in the same cluster)

Nor should they be in the same cluster simply because they are both linked to many different nodes as shown below.



(i and j should not be in the same cluster)

However, they should belong to the same cluster if they are linked to each other and to many common nodes as shown below.



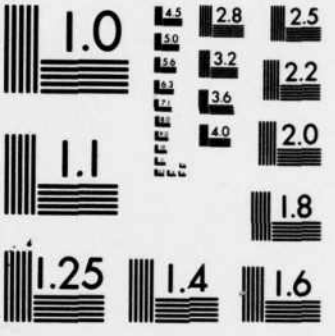
(i and j should be in the same cluster)

The above examples provide an intuitive "neighborhood" concept which corresponds to the nearest neighbor technique in statistical density estimation (see Chapter 6). To formalize the concept, the density contour between two nodes i,j (dc(i,j)) was defined in Section 7.2.1k. The level of the contour is used to identify the extent of linkage or connectedness between the nodes. Using this definition, the density contour dc(i,j) between the nodes i and j in the above three examples is 2/8,0, and 6/8 respectively. Further calculations are provided in detailed examples in Appendix A.

AD-A139 124 A CONSTRAINED-CLUSTERING APPROACH TO THE ANALYSIS OF REMOTE SENSING DATA(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH C J RYAN 1983 2/2

UNCLASSIFIED AFIT/CI/NR-83-80T F/G 12/1 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

The concepts above were necessary for the main definition describing the high-density clustering model which is related to the definition of a density contour cluster given in Section 7.2.1i.

Definition: A High-Density Cluster at level d^* on a graph G is a subset S such that S is maximal (contains the maximum number of nodes) among connected sets of nodes whose elements are connected by links with density contour $(dc(\cdot,\cdot)) \geq d^*$ (see Wong and Lattin, 1982).

It is easy to see from the preliminaries provided that the family T of high-density clusters on a graph forms a tree in that $A \in T$, $B \in T$ implies either A is contained in B, B is contained in A, or their intersection is empty. Since the level d^* of a high-density cluster identifies its degree of internal binding, the highly internally-connected subsets of the graph can be identified from the tree of high-density clusters defined on it. Examples of such clusters and trees are included in Appendix A.

As the density level d^* is reduced, the subgraph S at level d^* gradually expands until at some splitting level (going in reverse), d^*_s, it combines with other subgraphs to form a much larger subgraph. Such clusters which can not be expanded smoothly are called branching clusters and are important in determining the structure of the corresponding tree. They signify the number of modal (natural or high-density) subgraphs of the given graph. Then, a high-density cluster S is a branching cluster if every cluster properly including it contains a cluster disjoint from S. These branching high-density clusters form a tree that is a subset of the tree of all high-density clusters.

The high-density clustering model description is complete. The only tasks remaining are to provide an efficient implementing algorithm, to test this algorithm on actual graphs (Appendix A), and to apply it to the two data sets.

7.5 The High-Density Clustering Algorithm:

7.5.1 The Goals:

The requirement imposed on the defined high-density clustering model and its implementing algorithm have been implicitly stated earlier and will be explicitly given here. They are:

(i) it should partition a graph into well-connected (coherent, densely connected) subgraphs with minimal or relatively weak links between them,

(ii) it should require no prior knowledge of the number of disjoint, densely-connected subgraphs present,

(iii) it should be computationally efficient so that large graph networks can be effectively partitioned.

7.5.2 The Algorithm:

Given a set of N nodes numbered 1, 2, ... N with density contour $dc(i,j)$ on a graph G, the algorithm for finding the tree of high-density clusters follows:

(1) Let i and j be the pair of nodes with the densest link, amalgamate them to form a cluster C and define the density contour between that cluster and any node k by $dc(C,k) = \max (dc(i,k), dc(j,k))$,

(2) Repeat step (1), treating C as a node and ignoring i and j as separate entities. The amalgamation continues until all nodes are grouped into one larger cluster.

All clusters obtained in the course of this algorithm are high-density clusters. Note the similarity between

this algorithm and the single linkage joining algorithm given in the second paragraph of Section 7.3.3. Moreover, if step (1) above is replaced by "(1) begin with any node i and find node j with the densest link to node i, amalgamate them...", and (2) is replaced with "(2) Repeat step (1) treating C as the new starting node i and ignoring the original nodes i and j ...", it is easy to see this algorithm is equivalent to the classical minimum spanning tree (MST) algorithm provided in section 7.3.3(iv).

The MST algorithm was the one actually used in the implementing program (Appendix B) as seen in the examples of Appendix A.

The final step involves extracting the correct partitions from the tree of high-density clusters.

(3) Determine the number of minimal branching clusters. These are the high-density clusters desired. A branching cluster is minimal if it does not properly contain another branching cluster.

Note: for leftover nodes, Wong and Lattin (1982) have provided rules (not discussed here) for assigning them to the minimal branching clusters. For the purpose of evaluating this method and comparing it to the previous two, the leftover node assignments are not particularly relevant and will remain separated from the high-density clusters.

7.5.3 Comments on the Model and Algorithm:

The first step is to evaluate how well the goals of Section 7.5.1 have been met. Goal (i) is qualitative in that the optimal partitioning of the graph (system) should be one where each subgraph (subsystem) possesses high internal-binding (densely-connected) and simultaneously for each such subgraph to be weakly interconnected with the

the others. In fact, the model is defined with this in mind, and the examples in Appendix A indicate the model has certainly succeeded (for these examples at least).

Goal (ii) requires no prior knowledge of the number k of disjoint, densely-connected subgraphs. The proposed algorithm is independent of k and in fact determines k as explained above.

Goal (iii) is also qualitative and much harder to evaluate. An algorithm can be considered computationally efficient relative to: similar algorithms, theoretical minimums, memory space required, and many other criteria.

As previously mentioned, the MST algorithm is bounded by $0(N^2)$, but it is conjectured because of symmetry and the many recomputations of distance (density contours) required while updating, that it can be done in 0(n log n) (see Wong and Lattin, 1982). However, this had yet to be achieved.

In addition, the classical tradeoffs between the amoung of required memory, the complexity of the programming, and the number of calculations had to be addressed. The programming becomes simpler and faster if the full n by n distance (adjacency) matrix and n by n density contour matrix are used. For larger graphs (systems) with many nodes (basic elements) typically encountered in practice, the amount of required memory quickly surpasses available machine memory. In fact, such a program was written and successfully tested only on smaller graphs. But to handle larger systems, a program (see Appendix B) was written to take advantage of the symmetry of the problem, and sparcity of the matrices. A few specifics are further described in the examples of Appendix A. Further memory savings can be easily realized by additional symmetry considerations, but only at the expense of greatly increasing costly function

calls to the Min and Max Fortran functions. This idea was abandoned due to the time and overhead required of such calls. As written, the program is general in that it accepts either weighted or unweighted links between nodes. For unweighted links, a separate program could be written that did not require the weights at the links as input (all weights = 1), thus saving memory and processing time. Having addressed and basically met the goals (no claim is made to having found the most efficient algorithm), a few other comments are in order.

It is clear from the preliminaries of Sections 7.2 and 7.3 that the underlying ultrametric could be taken as the density contour values.

In order to better relate to the earlier MST and single linkage algorithms, there is a need for some measure of distance. This is easy to do with a suitable function of the density contours. For example, since a high density contour implies strong correctedness (strong interdependency), the distance between two nodes i and j could be defined as $\text{dist}(i,j) = \frac{1}{dc(i,j) + \text{constant}}$ where constant is added to prevent division by zero. In fact, this was done in the program (Appendix B).

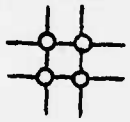
Another point brought out in the discussion of the single linkage method of Section 7.3.3 (i) was that of continuity. By their nature, unweighted graphs can not have continuous ultrametrics since a link can have only integer values 0 and 1. The method presented does preserve continuity in the case of the weighted graphs, however. Also, the algorithm does allow weighted graphs, eliminating a deficiency of several of the available methods discussed in Sections 7.3.1 and 7.3.2.

As a final comment, there is a minor deficiency in the model's algorithm as implemented. No special provision

has been made for handling ties. That is, if for example two previously formed clusters have equal density contours on a third cluster, no provision is made for breaking ties or choosing in which order to amalgamate them to the third cluster. Instead, the algorithm arbitrarily amalgamates one of the first two with the third, and then amalgamates all three together. In view of our goal of identifying the minimal branching clusters as high-density clusters, the deficiency presents no real problem (other than one of aesthetics). The three identified clusters would be disjoint high-density clusters within the set of a total of k such clusters. If further considerations require k to be smaller, i.e., the analyst desires fewer subsystems (clusters) for some reason, the three are combined into one larger cluster unless the analyst has overriding reasons for combining only two of the three (which could easily happen but still presents no problem). The third example of Appendix A illustrates this situation.

7.6 Application of the Graph Decomposition Method: In this section the graph decomposition method based on the high-density clustering model described in this chapter is applied to the two data sets to solve the constrained clustering problem. As with the previous two methods described in Chapters 5 and 6, the goal is to group (cluster) contiguous batches of pixels with similar multispectral measurements. As before, these clusters would hopefully represent the various agricultural fields found in the region covered by each particular data set.

The first step in applying this method to the data sets is the preliminary task mentioned in Section 7.1, i.e., to determine the basic elements of the system. Here, the basic elements are the pixels.

The second step is to decide if any interaction exists between each pair of elements (pixels) and, if so, to assign weights to each relationship based on the perceived degree of interaction. This step is certainly subjective. It requires intuition and judgment based on the particular situation to be analyzed. Here, the interaction is the geographical relationship between pixels. A somewhat arbitrary, but certainly not unreasonable, approach would be to link each pixel with the pixels directly above, below, and to either side of it. The linked graph would then have a pattern of the form  . Other schemes such as adding diagonal links to form the pattern  are certainly sensible, but the simpler pattern was chosen because it required less memory and adequately represented the physical situation.

Once the linkage pattern was chosen, weights had to be assigned. This was accomplished by a much more objective method. The weight between linked pixels i and j was defined as wt(i,j) = 1/(1+ss), where ss (sum of squares) was the sum of the squared differences between the four multispectral measurements of the two pixels. Therefore, for the geographically constrained clustering problem, whether or not two pixels were linked was a subjective decision which had to be made, while the weights of these links were assigned in a more objective, natural, way, i.e., as a function of the distance between their multispectral measurements. The result was a weighted, undirected graph.

The final step is to partition this graph into subgraphs (clusters of pixels) by applying the graph decomposition method of this chapter.

7.6.1 Data Set 1 Results: Figure 30 shows the resulting constrained clusters for Data Set 1. In this figure, clusters were formed if the density contour value between pixel pairs was greater than or equal to d^*. Here d^* was set equal to .0229. The choice of d^* is somewhat arbitrary. That is, the choice of d^* is determined by the analyst with due consideration given to the situation. If the value of d^* is chosen too large (as was the case in Figure 30), too many clusters are formed and too many pixels are in the "leftover node" category (pixels not assigned a cluster letter). As Figure 30 shows, there were 57 clusters formed at $d^* = .0229$ and over one fourth (111) of the pixels were "leftover".

If, on the other hand, d^* is chosen sufficiently small (.001 for example) there would be no "leftover" pixels and only one cluster. Thus, all the information contained in the graph is lost.

Figure 31 shows the resulting constrained clusters for $d^* = .0169$. It shows 22 clusters and 53 "leftover" pixels for a decrease of more than 50% in each category.

A comparison of Figure 2 and 31 reveals the accuracy of the method. Cluster A in Figure 31 represents the red (hatched) area in Figure 2. However, there are several "leftover" pixels in and around this area. These pixels in the red area are 2(.0152), 3(.0133), 8(.0160), 49(.0107), 81(.0151), and 82(.0151). The numbers in parentheses are the values of d^* at which these pixels would have been amalgamated into Cluster A. All of these density contour values (except possibly that for 49) are relatively high. In contrast, for example, those outside the area include pixels 9(.0053), 29(.0053), and 70(.0065). Their amalgamation levels into Cluster A are very small relative to those of the previous group inside the red area. In

Figure 30

Data Set 1

($d^* = .0229$)

			4 B	5	6 D	7 D	8	9	10	11 Z1	12 Z1	13 Z1	14 Z1	15 Z1	16 C2	17 C2	18 C2	19 C2	20 C2
21	22	23	24 B	25 D	26 D	27 D	28 D	29	30 Z1	31 Z1	32 Z1	33 Z1	34 Z1	35 Z1	36	37 C2	38 C2	39 C2	40 C2
41 A	42 A	43 C	44 C	45 D	46 D	47 D	48 D	49	50 Z1	51 Z1	52 Z1	53 Z1	54	55 M	56	57	58 M1	59 M1	60 M1
61 A	62 A	63 A	64 C	65 C	66 D	67 D	68 D	69	70	71 E	72 H	73 H	74 H	75 M	76 I1	77	78 L1	79 L1	80
81	82	83	84	85	86 E2	87 E2	88	89	90 E	91 E	92 F	93 H	94	95 M	96 I1	97	98 L1	99 L1	100
101 B2	102 B2	103	104 D2	105 D2	106	107 G	108 G	109 E	110 E	111 E	112 F	113 H	114 H	115	116	117 L1	118 L1	119 L1	120 L1
121	122	123 N	124 U	125 U	126 J	127 J	128 J	129 J	130	131 I	132 I	133 L	134 L	135 L	136 K1	137 K1	138 L1	139 L1	140 L1
141	142	143 N	144 N	145	146 J	147 K	148 K	149	150	151	152 X1	153 X1	154 J1	155 J1	156 J1	157	158	159 N1	160 N1
161	162 N	163 N	164 N	165 N	166	167 C1	168 C1	169 C1	170	171 A2	172 Y1	173 Y1	174	175	176	177 O1	178 O1	179 O1	180 S1
181 N	182 N	183 N	184 N	185 N	186 N	187 N	188	189	190	191 A2	192 A2	193 A2	194	195	196	197 O1	198 P1	199 O1	200 S1
201	202	203 P	204 P	205	206	207 N	208	209	210	211	212	213	214	215	216 P1	217 O1	218 P1	219 R1	220 R1
221 S	222 S	223 P	224 O	225 O	226 N	227 N	228	229 E1	230 F1	231 F1	232 F1	233 F1	234 F1	235 P1	236 P1	237 P1	238 P1	239	240 R1
241 D1	242 D1	243 D1	244 D1	245 O	246 O	247	248	249 E1	250 F1	251 F1	252 F1	253 F1	254 G1	255	256	257 P1	258 P1	259 Q1	260 Q1
261 D1	262 D1	263 Q	264 Q	265 O	266	267	268 R	269 H1	270 H1	271 F1	272 F1	273 G1	274 G1	275	276	277	278	279	280
281 D1	282	283 V	284 Q	285 O	286 R	287 R	288 R	289 R	290 R	291 R	292 R	293 A1	294 A1	295	296 T1	297 T1	298	299 U1	300 U1
301 V	302 V	303 V	304 V	305 V	306	307 R	308 R	309 R	310 R	311 R	312 T	313 A1	314 B1	315 B1	316	317 T1	318 T1	319 T1	320 U1
321	322 V	323 V	324	325 Y	326 Y	327 Y	328 R	329	330 Y	331	332 T	333 A1	334 A1	335 B1	336	337	338	339 T1	340 T1
341 W	342 W	343 W	344	345 X	346 Y	347 Y	348 Y	349 Y	350 Y	351	352 Z	353 Z	354 A1	355 A1	356	357	358 T1	359 T1	360 T1
361 W	362 W	363 W	364	365 X	366	367 Y	368 Y	369 Y	370 Y	371 Y	372 Z	373	374 A1	375 A1	376	377 V1	378 T1	379 T1	380 T1
381 W	382 W	383 W	384	385 X	386	387	388 Y	389	390 Y	391 Y	392 Y	393	394	395	396	397 V1	398 T1	399 W1	400 W1

Note: letters or letters followed by numbers represent clusters

Figure 31

Data Set 1

($d^* = .0169$)

1 A	2	3	4 A	5 A	6 A	7 A	8	9	10 P	11 P	12 P	13 P	14 P	15 P	16 S	17 S	18 S	19 S	20 S
21 A	22 A	23 A	24 A	25 A	26 A	27 A	28 A	29	30 P	31 P	32 P	33 P	34 P	35 P	36 S	37 S	38 S	39 S	40 S
41 A	42 A	43 A	44 A	45 A	46 A	47 A	48 A	49	50 P	51 P	52 P	53 P	54 B	55 B	56 J	57 J	58 J	59 J	60 J
61 A	62 A	63 A	64 A	65 A	66 A	67 A	68 A	69 A	70	71 B	72 B	73 B	74 B	75 B	76 G	77 J	78 J	79 J	80
81	82	83	84	85	86 V	87 V	88 B	89 B	90 B	91 B	92 B	93 B	94 B	95 B	96 G	97	98 J	99 J	100 J
101 R	102 R	103	104 U	105 U	106 B	107 B	108 B	109 B	110 B	111 B	112 B	113 B	114 B	115 B	116 G	117 J	118 J	119 J	120 J
121 R	122	123 C	124 C	125 C	126 B	127 B	128 B	129 B	130 B	131 B	132 B	133 B	134 B	135 B	136 I	137 I	138 J	139 J	140 J
141	142	143 C	144 C	145 C	146 B	147 B	148 B	149 B	150 B	151 B	152 O	153 O	154 H	155 H	156 H	157 H	158 K	159 K	160 K
161 C	162 C	163 C	164 C	165 C	166 B	167 D	168 D	169 D	170 D	171 Q	172 O	173 O	174	175 M	176 M	177 L	178 L	179 L	180 M
181 C	182 C	183 C	184 C	185 C	186 C	187 C	188 D	189 D	190	191 Q	192 Q	193 Q	194	195	196 M	197 L	198 M	199 L	200 M
201 C	202	203 C	204 C	205 C	206	207 C	208	209 F	210	211 Q	212	213	214	215 M	216 M	217 L	218 M	219 M	220 M
221 C	222 C	223 C	224 C	225 C	226 C	227 C	228 F	229 F	230 F	231 F	232 F	233 F	234 F	235 M	236 M	237 M	238 M	239 M	240 M
241 E	242 E	243 E	244 E	245 C	246 C	247	248 C	249 F	250 F	251 F	252 F	253 F	254 F	255 T	256	257 M	258 M	259 M	260 M
261 E	262 E	263 C	264 C	265 C	266 C	267 C	268 C	269 F	270 F	271 F	272 F	273 F	274 F	275 T	276 T	277	278 M	279 M	280
281 E	282	283 C	284 C	285 C	286 C	287 C	288 C	289 C	290 C	291 C	292 C	293 C	294 C	295 T	296 N	297 N	298 N	299 N	300 N
301 C	302 C	303 C	304 C	305 C	306	307 C	308 C	309 C	310 C	311 C	312 C	313 C	314 C	315 C	316 C	317 N	318 N	319 N	320 N
321	322 C	323 C	324	325 C	326 C	327 C	328 C	329	330 C	331	332 C	333 C	334 C	335 C	336 C	337	338	339 N	340 N
341 C	342 C	343 C	344 C	345 C	346 C	347 C	348 C	349 C	350 C	351	352 C	353 C	354 C	355 C	356 C	357 N	358 N	359 N	360 N
361 C	362 C	363 C	364	365 C	366	367 C	368 C	369 C	370 C	371 C	372 C	373	374 C	375 C	376	377 N	378 N	379 N	380 N
381 C	382 C	383 C	384 C	385 C	386	387	388 C	389	390 C	391 C	392 C	393	394	395	396	397 N	398 N	399 N	400 N

Note: letters represent clusters

other words, for a choice of $d^* = .0107$, Figures 2 and 31 would have a perfect match in the red area. It appears, at least from the results for this data set, the graph decomposition technique provides accurate, constrained clusters.

7.6.2 Data Set 2 Results: Figure 32 shows the resulting constrained clusters formed for Data Set 2 when $d^* = .0095$. The density contour threshold level (d^*) was chosen lower than for Data Set 1 to show how the procedure performs at various d^* settings. The number of clusters is only a little smaller than for $d^* = .0169$ for Data Set 1 in Figure 31, but there are only 13 "leftover" pixels, as opposed to 53 in Figure 31.

An inspection of Figure 32 reveals there are several pixels that have been clustered contrary to the red, non-red patterns of Figure 3. Pixels 73, 287, 288, and 289 are considered red but should not be. Pixels 88 and 89 are considered non-red, but should be red. These are all borderline pixels in the sense described earlier. Pixels 376-389 and 392-400 are also considered red but should not be. Figure 1 shows these pixels to be a light shade of "gray" as are the red pixels. However, the colored photograph shows this set of pixels to be white (or light) and not red. Apparently they are close enough spectrally to be combined with the red pixels at this density contour threshold value. Therefore, $d^* = .0095$ appears too low.

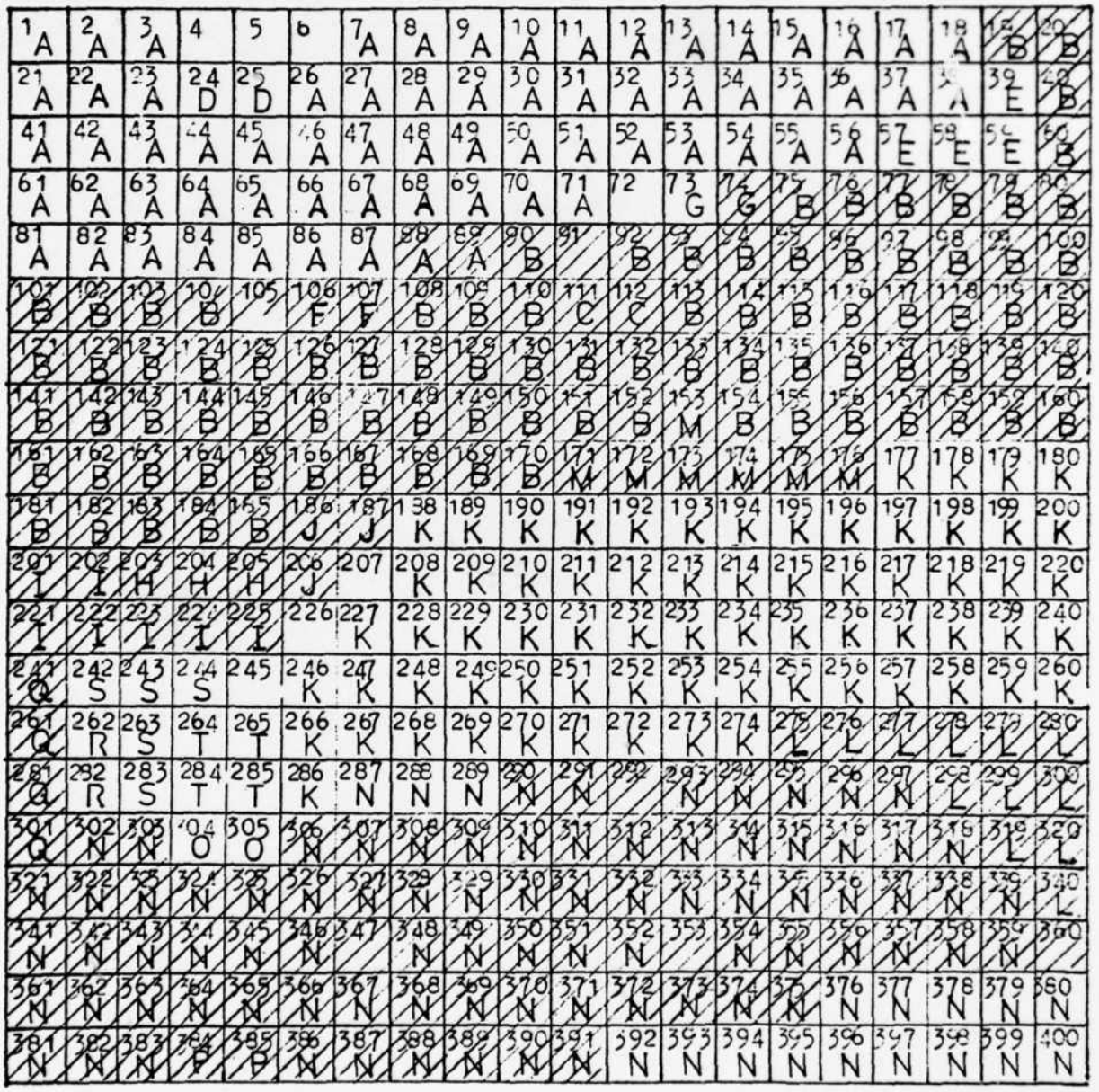
Figure 33 shows the constrained clusters formed in Data Set 2 for $d^* = .0177$. There are many more clusters and "leftover" pixels here than in Figure 32, but this was expected since d^* is much larger in Figure 33.

A comparison of Figures 3 and 33 reveals the clustering assignments to be much more accurate than those of Figure 32.

Figure 32

Data Set 2

($d^* = .0095$)

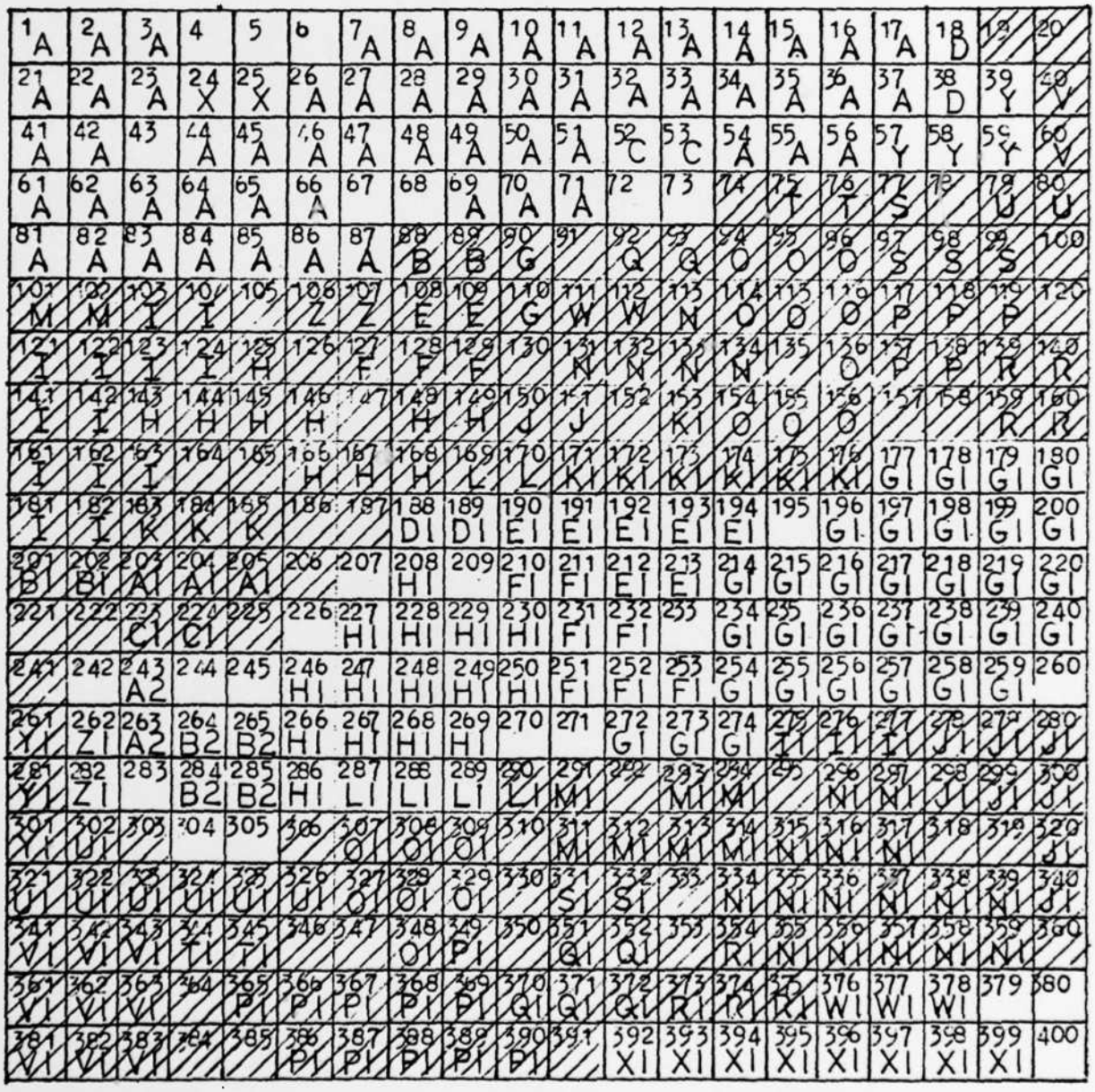


Note: letters represent clusters

Figure 33

Data Set 2

($d^* = .0177$)



Note: letters or letters followed numbers represent clusters

Only pixel 290 was misclassified (misclustered). It was in a non-red cluster while Figure 3 shows it to be red. Again, it is a borderline pixel in the sense described earlier, and the method should have difficulty clustering it correctly. The "leftover" pixels within the red area of Figure 3 but near the non-red area include 20(.0116), 74(.0168), 78(.0133), 186(.0168), 187(.0082), 206(.0133), 222(.0107), 225(.008), 241(.0176), 303(.0127), and 391(.0113). Again the numbers in parentheses are the values of d^* at which these pixels would have been included in the red area. With the exceptions of pixels 187 and 225 these values are relatively high. "Leftover" pixels outside the red area of Figure 3 include 19(.0093), 72(.006), 73(.006), 207(.0093), 226(.0082), 242(.0067), 244(.0107), 245(.0065), 260(.0093), 304(.0125), and 305(.0091). With the exception of 244 and 304, their threshold density contour values are relatively very low.

As a result, for a $d^* = .01$, only five pixels would be misclustered according to Figure 3, i.e., pixels 187, 225, 244, 290, and 304. Each of these is obviously a borderline case.

For Data Set 2, as with Data Set 1, the conclusion is that this graph decomposition method performs very well in partitioning the graph into the correct, constrained clusters.

7.7 Method Evaluation and Problems: In this chpater, a new, computationally efficient algorithm has been described, which partitions a graph (weighted or unweighted) into strongly coherent subgraphs (subsystems) which are separated from each other by relatively few or relatively weak links. Based on the high-density cluster model defined in Section 7.4, the implementing algorithm extracts the appropriate partition

from the tree of high-density clusters defined in Section 7.2.

Several examples are provided in Appendix B to verify the model and the algorithm's effectiveness. Actually, over 20 examples were tested, all successfully. In addition, the method was successfully applied to both data sets.

The high-density clustering concept developed by Wong and Lattin (1982) appears to be a useful contribution to the general graph decomposition methodology and is related to other existing methods as described in Section 7.3.

The goals set for the method appear to have been met, and the applications support this belief. The minor, nagging question of ties was shown to present no real problems for an analyst and does not detract from the considerable worth of the algorithm. In any case, it could easily be removed by a slight adjustment to the implementing algorithm, but was not considered worth the effort.

Several problems or weaknesses in this decomposition method surfaced during the technique description and application. Figure 33, when compared with Figure 3, revealed several pixels were clustered into "incorrect" areas, i.e., non-red clusters were amalgamated into red clusters and vice versa. As with the previous methods of Chapters 5 and 6, this method had trouble with borderline pixels, i.e., pixels whose subjective classifications into red and non-red categories were dubious at the outset. "Leftover nodes" (pixels) fill in this same borderline category. Therefore, the method performed as well as could possibly be expected under the circumstances.

There are, however, several weaknesses associated with this method. First, although the number K of clusters is not required by the algorithm, the method does not automatically provide this number upon completion. Instead, the analyst had to use knowledge of the situation and some judgment in determining the proper level at which

to set the density contour threshold d^* in order to form the clusters and thus determine k. As with the k^{th} nearest neighbor algorithm, the method does aid the analyst in determining the number of clusters by providing the tree of high-density clusters. This aid turns out to be very useful for the graph decomposition method.

Second, the analyst must make subjective determinations regarding the independence or interdependence between pairs of elements in the system and the level of interdependence (the proper weight for the link) when it exists. This problem arose mostly because the constrained clustering problem was converted to a graph decomposition problem. This weakness, then, is unavoidable and worthwhile in light of the power and usefulness of the technique.

Finally, the "leftover" nodes proved troublesome. They were a definite weakness of the algorithm because various pixels were not assigned to clusters for some sufficiently high value of d^*. However, it must be stressed that this is only a weakness in the implementing algorithm, deliberately chosen to simplify the programming. The actual technique developed by Wong and Lattin (1982) provides the necessary set of rules for handling all "leftover nodes".

None of the above weaknesses detract from, to any real extent, the usefulness of the technique in attacking the geographically constrained clustering problem for our multispectral data.

Chapter 8

Summary and Conclusions

8.1 Summary: Due to great strides in electronic technology and an accompanying decrease in the cost of sophisticated hardware, the earth's resources as well as man made objects are being extensively surveyed by airborne and spaceborne sensors. As a result, there is a plethora of imagery data available for analysis. One example is the Landsat II multispectral imagery data for a variety of agricultural crops and vegetative backgrounds. This data set was extensively analyzed by Misra and Wheeler (1978, 1980) in an attempt to differentiate between crops and to separate crops from other forms of vegetation present. Their goal was to characterize crops by their "signatures" on the four wavelengths measured, resulting in inventories of the various crops.

A number of physical complications made Misra and Wheeler's task very difficult. Factors such as incomplete data due to atmospheric conditions when the satellite passed over the site; different crop calendars, planting dates, climates, and soil treatments; and the very different physical appearances of a given crop for different phases of growth within their calendars all combined to make a crop's multispectral signature very complex and illusive. It is apparent that a considerable amount of research remains before their goals can be completely met and crops can be identified reliably.

With a valuable perspective gained by Misra and Wheeler's work, this study retained the assumption that different crops appear physically different, and therefore, have different multispectral measurements. The goal of this study was to evaluate various methods of separating different crops found in a single ground image.

Three methods of clustering like objects were evaluated as to their performance in separating different Landsat scenes. These methods were the K-Means Clustering Method, the K^{th} Nearest Neighbor Clustering Method, and the Graph Decomposition Method.

Prior to applying and evaluating these methods, a preliminary analysis was performed on two subsets of one LACIE image acquisition to provide a foundation for understanding the nature of the data to be analyzed. The findings of this preliminary analysis verified Misra and Wheeler's conclusions that the four dimensional data was essentially two dimensional, and that the within field variability of the measurements was too large to hope for simple crop classification signatures. The importance of these two findings was that the four-dimensional data could be graphically displayed without much loss of information, and that geographical considerations would facilitate identification of field boundaries.

With these results providing the proper framework within which to evaluate the three clustering methods, the methods were applied to the two data subsets.

All three of the methods were very successful in their overall task of identifying contiguous batches of pixels with similar multispectral measurements. Each had problems with borderline or boundary pixels that provided a transaction between two areas of contrasting shades of gray. The particular strengths and weaknesses of each method are provided in the following section.

8.2 Strengths and Weaknesses of the Three Clustering Methods:

8.2.1 Evaluation of the K-Means Clustering Method:

Weaknesses of this method include determining the value

of K beforehand, the method's assumptions, and the resulting locally (as opposed to globally) optimal solutions which are dependent upon scale and initial partitions. The underlying model's assumptions include a search for spherical clusters. Also, the method fails to consider geographical information when forming clusters, so that spatial clustering must be done by hand.

Its strengths include the generality of its application, acceptable results, availability of implementing programs, and its properties have been well studied and are readily available.

8.2.2 Evaluation of the K^{th} Nearest Neighbor Clustering Method: Weaknesses of this method include assignment of "leftover" pixels and the dependency of cluster membership on the chosen value of K. The final choice of K remains an unsolved problem. As with K-Means, this method fails to consider the geographical information present and spatial clustering must be done by hand.

Its strengths include the acceptability of the resulting cluster assignments, the efficiency of its implementing algorithm, and more importantly, the diagnostic plots for estimating the number of modes (clusters) in the underlying population distribution.

8.2.3 Evaluation of the Graph Decomposition Method: Weaknesses of this method include assignment of "leftover" pixels as with the K^{th} Nearest Neighbor method. Although the number of clusters is not a required input for the algorithm as it is with the K-means method, the method does not automatically provide this number as output. Instead, the analyst must choose the density contour threshold value based on intuition and prior knowledge of the situation. Only then is the number of clusters determined.

Its strengths include aiding in the determination of the number of clusters by providing the tree of high-density clusters, and the method does not require the number of clusters as input. Wong and Lattin (1982) provide a comprehensive set of rules for efficiently assigning "leftover" nodes to the existing clusters. Finally, unlike the other two methods, this method considers the geographical information contained in the data when forming the high-density clusters using a computationally efficient algorithm which takes advantage of the symmetry involved.

8.3 Conclusions: The work of Misra and Wheeler (1978, 1980) went a long way toward meeting the goal of providing crop product inventories over large areas of land. However, the multitude of physical complications they encountered made their goal a formidable one. Much work remains before crop signatures, temporal trajectories, or similar spectral attributes of a crop are sufficiently characterized to meet their goal.

Their work did provide several useful results. The fact that the four-channel data was essentially two dimensional (with two of the wave bands linearly dependent on the others) lead to the ability to graphically display the data. Perhaps future spaceborne sensors will only be required to collect two dimensional information for this application, or data could be collected at other more carefully considered wavelengths to glean the crucial information necessary to distinguish between crops. More importantly, their work identified the costs of such an undertaking and provided a better perspective of the physical obstacles to be overcome. This perspective provided motivation for limiting the scope of the present study to data from a single point in time.

The seemingly more limited goal of identifying spectrally similar data from a single area acquisition led to the evaluation of three possible clustering methods designed to provide the desired results. While the methods did not attempt to identify individual crops, they did accurately separate the different fields present in the areas examined. In addition, the application of these methods is far more general than the application specific work of Misra and Wheeler. These methods can each be applied to a more general set of situations requiring clustering in both the spectral measurement space, and geographical or spatial groupings. Such double clustering was referred to as the "constrained-clustering" problem.

The "constrained-clustering" problem can be applied to many other totally unrelated situations such as decomposing complex software design problems into more manageable subproblems, and in exploring the structure of social networks as discussed in Wong and Lattin (1982).

Which of the three methods to use for a particular application can only be determined based on what is already decided a priori and the objectives of the project. The Graph Decomposition method is the only one of the three which considers geographical nearness information (or more generally the degree of interdependence between pairs of elements in the system). For this reason and because of its computational efficiency, it appears to have a slight edge over the other two methods. In any case, there is no denying the usefulness of the two new clustering methods when compared to the classical K-means method for solving "constrained-clustering" problems. Certain situations and objectives might even require two or all three methods be used to attain goals. The two new clustering methods can be utilized in a wide variety of useful applications such as crop identification in a "constrained-clustering" environment.

Appendix A

Examples Using the Graph Decomposition Method of Chapter 7

Having described (in Chapter 7) the graph decomposition method based on the high-density clustering model and an algorithm for implementing it, several examples are now provided to test these ideas and at the same time clarify them.

Example 1:

Figure A1 shows the unweighted graph presented in Section 7.4. The nodes have been numbered for identification and processing within the program, and the density contours of the links are provided in parentheses. The circles enclosing sets of nodes represent high-density clusters at level d^*, where d^* is also included for reference. Here there are two high-density clusters at level $d^* = .8$ and two slightly large clusters at $d^* = .67$. These are all high-density clusters and those at $d^* = .67$ are minimal branching clusters. At level $d^* = .25$ (a splitting level) the branching clusters expand into a single large cluster. Here $k = 2$ and the subgraphs desired are {1,2,3,4,5} and {6,7,8,9,10}. By hand computation, the model appears effective in this example.

Figure A2 shows the required input data (top) and the results (bottom half) for Example 1. The data requires an N by (M+1) adjacency matrix where N = the number of nodes (here 10) and M = the maximum number of links to a given node (here 4). The final column (last field of four digits) is required by the program to determine the number of links to node i where i is the row number. The remaining four columns provide the numbers of the nodes

Figure A1

Graph for Example 1

Example 1

Number of nodes = 10

Number of links = 17

Note: The density contours are in parentheses.

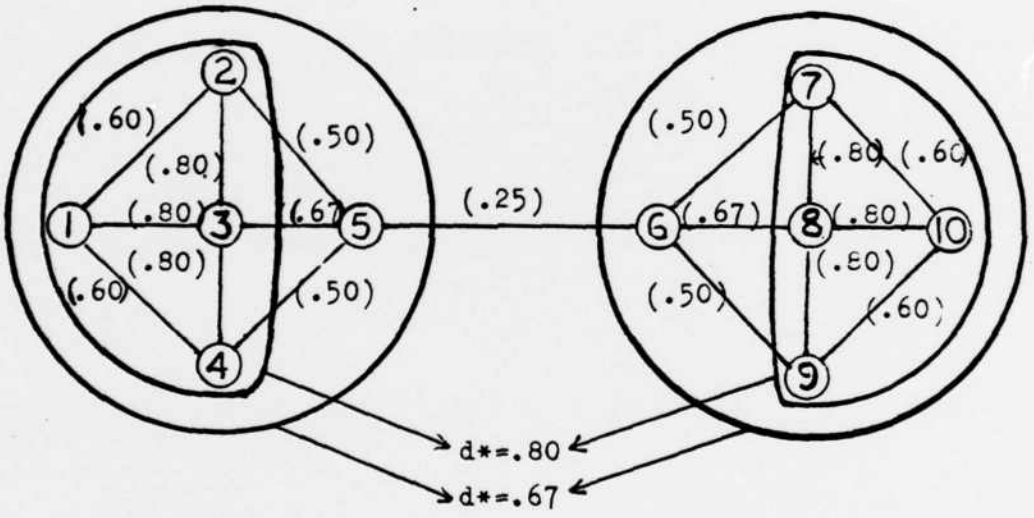


Figure A2

Example 1 Data and Results

Example 1 Data

```
00020003000400000003
00010003000500000003
00010002000400050004
00010003000500000003
00020003000400060004
00050007000800090004
00060008000100000003
00060007000900100004
00060008001000000003
00070008000900000003
1.0001.0001.0000.000
1.0001.0001.0000.000
1.0001.0001.0001.000
1.0001.0001.0000.000
1.0001.0001.0001.000
1.0001.0001.0001.000
1.0001.0001.0000.000
1.0001.0001.0001.000
1.0001.0001.0000.000
1.0001.0001.0000.000
```

Example 1 Results

THE MINIMUM SPANNING TREE

```
NODE  1 CONNECTED TO NODE  3 WITH DENSITY CONTOUR = 0.8000
NODE  3 CONNECTED TO NODE  2 WITH DENSITY CONTOUR = 0.8000
NODE  2 CONNECTED TO NODE  4 WITH DENSITY CONTOUR = 0.8000
NODE  4 CONNECTED TO NODE  5 WITH DENSITY CONTOUR = 0.6667
NODE  5 CONNECTED TO NODE  6 WITH DENSITY CONTOUR = 0.2500
NODE  6 CONNECTED TO NODE  8 WITH DENSITY CONTOUR = 0.6667
NODE  8 CONNECTED TO NODE  7 WITH DENSITY CONTOUR = 0.8000
NODE  7 CONNECTED TO NODE  9 WITH DENSITY CONTOUR = 0.8000
NODE  9 CONNECTED TO NODE 10 WITH DENSITY CONTOUR = 0.8000
NODE 10 CONNECTED TO NODE  0 WITH DENSITY CONTOUR = 0.0000
```

THE CLUSTER TREE

```
NODE  1 DC 0.8000 ____
NODE  3 DC 0.8000 |
NODE  2 DC 0.8000 |
NODE  4 DC 0.6667 |___ ____
NODE  5 DC 0.2500 |    |
NODE  6 DC 0.6667 |____|
NODE  8 DC 0.8000 |
NODE  7 DC 0.8000 |
NODE  9 DC 0.8000 |
NODE 10 DC 0.0000 |____|____|____|
```

connected to node i, e.g., node 1 is connected to nodes 2, 3, and 4 for a total of 3 (last column) links. The second matrix directly below the first is N by M and contains the values of the links. In this case, since the graph is unweighted, all values are 1. As mentioned previously, in these cases the matrix is not required. The general program could easily be modified to run without it, thus saving memory and processing time.

Below the data is the output of the program. The Minimum Spanning Tree (MST) shows the proper minimum distance paths identified in Figure A1 (recall that distance = 1/(dc + constant)). The information in the MST is sufficient to construct the high-density cluster tree shown below the MST. The tree correctly identifies the two desired high-density clusters. Note that in the tree, the dc value on a given line corresponds to the level at which the node on that line is connected to the node on the <u>following</u> line. For example, node 2 is connected to node 4 at density level .8 and node 9 is connected to node 10 at level .8, while node 5 is connected to node 6 at level .25. These values can be confirmed in the preceeding MST. These density levels could be placed between node lines for each of reading but would result in trees twice as long.

Example 2:

Figure A3 shows a weighted graph (density contours are in parentheses, weights are not). Again, the method correctly identifies the three high-density clusters. It is also clear which clusters would be chosen for $d^* = .2$ if only two clusters were desired. Figure A4 provides the input data and output results for this example. They are similar to those for Example 1 except for the weighted links

Figure A3

Graph for Example 2

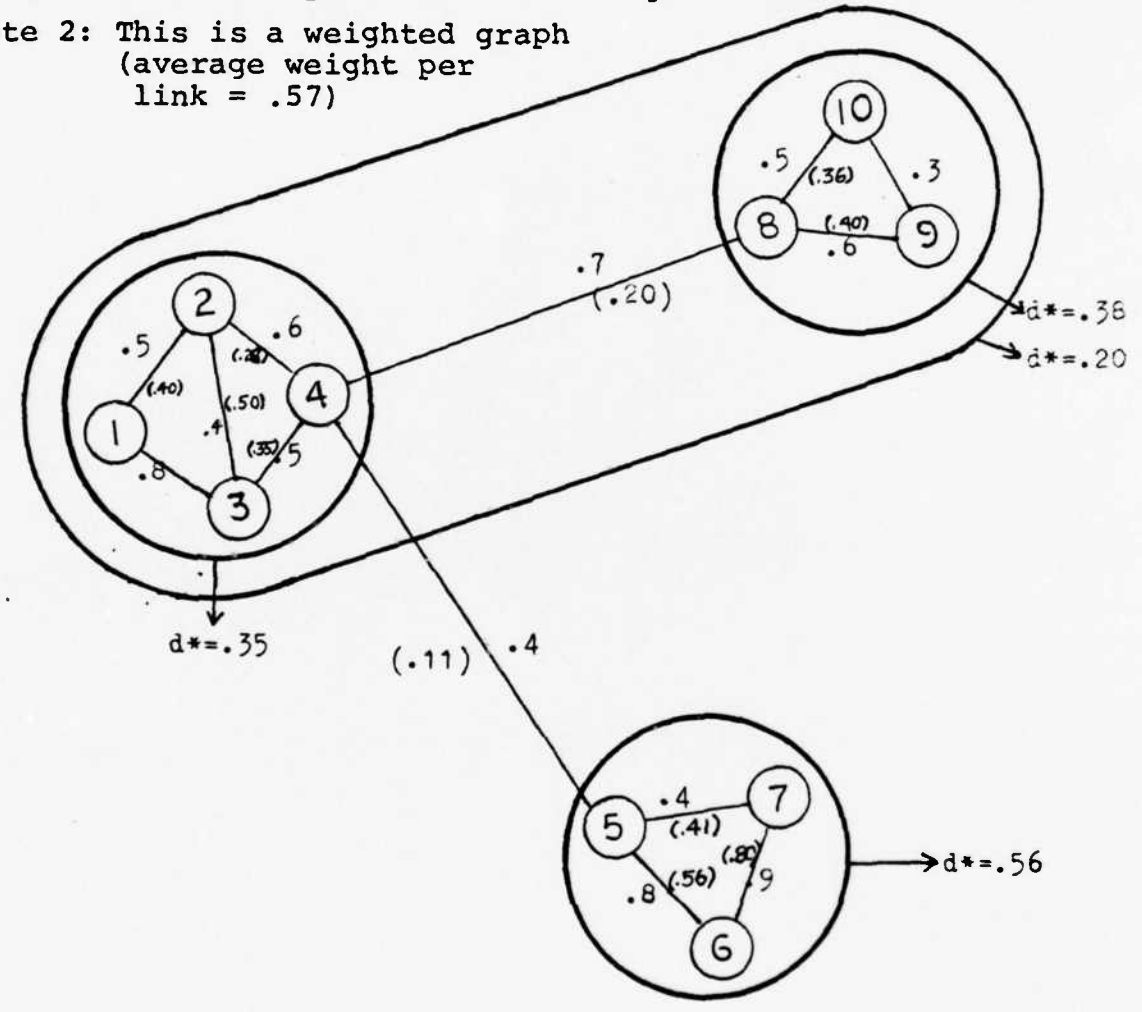
Example 2

Number of nodes = 10

Number of links = 13

Note 1: The density contours are in parentheses.

Note 2: This is a weighted graph (average weight per link = .57)



Three Branching Clusters; best partition is:

Subgraph 1: {1, 2, 3, 4}

Subgraph 2: {8, 9, 10}

Subgraph 3: {5, 6, 7}

Figure A4

Example 2 Data and Results

Example 2 Data

```
00020003000000000002
00010003000400000003
00010002000400000003
00020003000500000004
00040006000700000003
00050007000000000002
00050006000000000002
00040009001000000003
00080010000000000002
00080009000000000002
0.5000.8000.0000.000
0.5000.4000.6000.000
0.8000.4000.5000.000
0.6000.5000.4000.700
0.4000.8000.4000.000
0.8000.9000.0000.000
0.4000.9000.0000.000
0.7000.6000.5000.000
0.6000.3000.0000.000
0.5000.3000.0000.000
```

Example 2 Results

THE MINIMUM SPANNING TREE

```
NODE  1  CONNECTED TO NODE  3 WITH DENSITY CONTOUR = 0.5125
NODE  3  CONNECTED TO NODE  2 WITH DENSITY CONTOUR = 0.5000
NODE  2  CONNECTED TO NODE  4 WITH DENSITY CONTOUR = 0.2750
NODE  4  CONNECTED TO NODE  8 WITH DENSITY CONTOUR = 0.2000
NODE  8  CONNECTED TO NODE  9 WITH DENSITY CONTOUR = 0.4000
NODE  9  CONNECTED TO NODE 10 WITH DENSITY CONTOUR = 0.3833
NODE 10  CONNECTED TO NODE  5 WITH DENSITY CONTOUR = 0.1143
NODE  5  CONNECTED TO NODE  6 WITH DENSITY CONTOUR = 0.5625
NODE  6  CONNECTED TO NODE  7 WITH DENSITY CONTOUR = 0.8000
NODE  7  CONNECTED TO NODE  8 WITH DENSITY CONTOUR = 0.0000
```

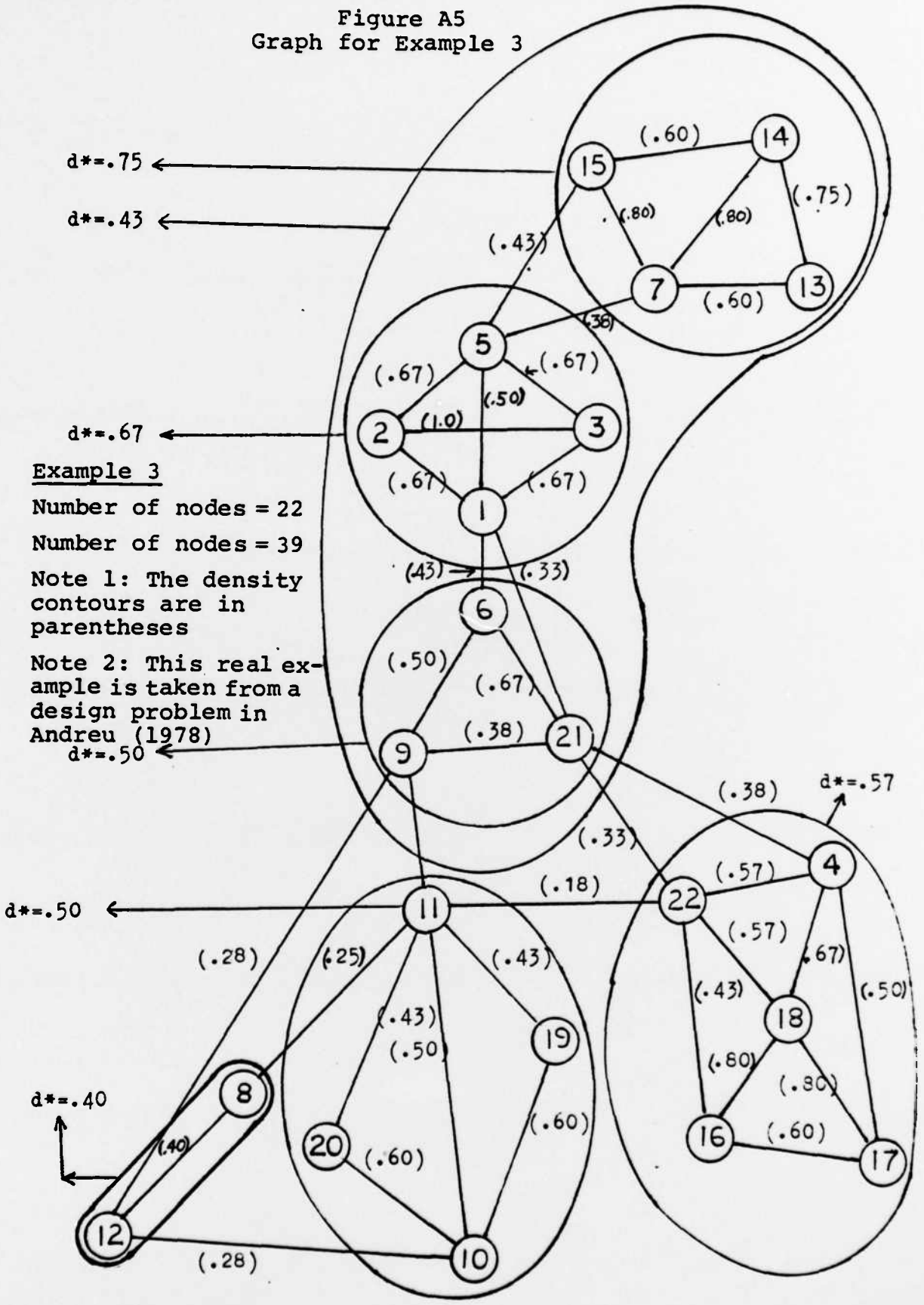
THE CLUSTER TREE

```
NODE  1 DC 0.5125
NODE  3 DC 0.5000
NODE  2 DC 0.2750
NODE  4 DC 0.2000
NODE  8 DC 0.4000
NODE  9 DC 0.3833
NODE 10 DC 0.1143
NODE  5 DC 0.5625
NODE  6 DC 0.8000
NODE  7 DC 0.0000
```

which are entered in the lower data matrix. The cluster tree again forms the proper clusters at the correct levels.
Example 3:

Figure A5 shows an unweighted graph for an actual (real) system. The system is a design problem where links indicate interdependencies among requirements. Here the number of high-density clusters (k) is 6 (alternatively k could be 4 by combining the top 3 clusters). This example illustrates a situation involving ties, i.e. the top 3 clusters are all connected at level .43 in Figure A5. However, the cluster tree in Figure A6 shows that clusters {6, 9, 21} and {7, 13, 14, 15} are first combined at level .43 and then cluster {1, 2, 3, 5} is joined to them at the same level. This would present no special problem to an analyst, provided the levels of expansion are examined. Note that this problem is less likely with weighted graphs. For completeness, Figure A7 shows the required input data for this example. Here N = 22 and M = 6, since there are 22 nodes and node 11 is linked to the most nodes, 6.

Figure A5
Graph for Example 3



Example 3

Number of nodes = 22

Number of nodes = 39

Note 1: The density contours are in parentheses

Note 2: This real example is taken from a design problem in Andreu (1978)

Figure A6

Example 3 Results

THE MINIMUM SPANNING TREE

```
NODE  1 CONNECTED TO NODE  2 WITH DENSITY CONTOUR = 0.6667
NODE  2 CONNECTED TO NODE  3 WITH DENSITY CONTOUR = 1.0000
NODE  3 CONNECTED TO NODE  5 WITH DENSITY CONTOUR = 0.6667
NODE  5 CONNECTED TO NODE  6 WITH DENSITY CONTOUR = 0.4286
NODE  6 CONNECTED TO NODE 21 WITH DENSITY CONTOUR = 0.6667
NODE 21 CONNECTED TO NODE  9 WITH DENSITY CONTOUR = 0.5000
NODE  9 CONNECTED TO NODE 15 WITH DENSITY CONTOUR = 0.4286
NODE 15 CONNECTED TO NODE  7 WITH DENSITY CONTOUR = 0.8000
NODE  7 CONNECTED TO NODE 14 WITH DENSITY CONTOUR = 0.8000
NODE 14 CONNECTED TO NODE 13 WITH DENSITY CONTOUR = 0.7500
NODE 13 CONNECTED TO NODE  4 WITH DENSITY CONTOUR = 0.3750
NODE  4 CONNECTED TO NODE 18 WITH DENSITY CONTOUR = 0.6667
NODE 18 CONNECTED TO NODE 16 WITH DENSITY CONTOUR = 0.8000
NODE 16 CONNECTED TO NODE 17 WITH DENSITY CONTOUR = 0.8000
NODE 17 CONNECTED TO NODE 22 WITH DENSITY CONTOUR = 0.5714
NODE 22 CONNECTED TO NODE 12 WITH DENSITY CONTOUR = 0.2857
NODE 12 CONNECTED TO NODE  8 WITH DENSITY CONTOUR = 0.4000
NODE  8 CONNECTED TO NODE 10 WITH DENSITY CONTOUR = 0.2857
NODE 10 CONNECTED TO NODE 19 WITH DENSITY CONTOUR = 0.6000
NODE 19 CONNECTED TO NODE 20 WITH DENSITY CONTOUR = 0.6000
NODE 20 CONNECTED TO NODE 11 WITH DENSITY CONTOUR = 0.5000
NODE 11 CONNECTED TO NODE  0 WITH DENSITY CONTOUR = 0.0000
```

THE CLUSTER TREE

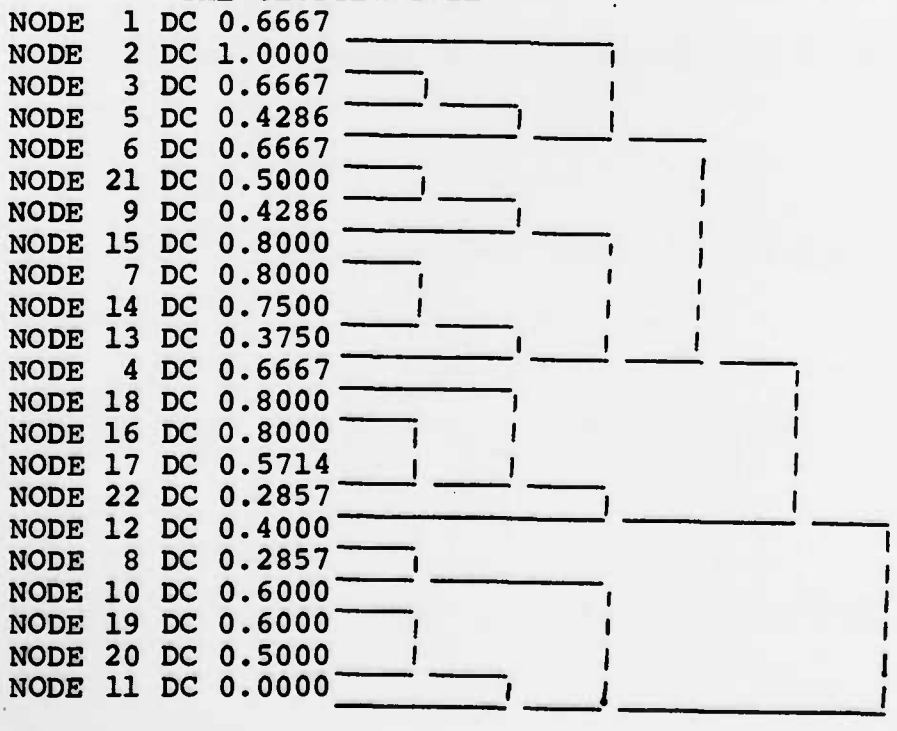


Figure A7

Example 3 Data

```
00020003000500060021000000005
00010003000500000000000000003
00010002000500000000000000003
00170018002100220000000000004
00010002000300070015000000005
00010009002100000000000000003
00050013001400150000000000004
00110012000000000000000000002
00060011001200210000000000004
00110012001900200000000000004
00080009001000190020002200006
00080009001000000000000000003
00070014000000000000000000002
00070013001500000000000000003
00050007001400000000000000003
00170018002200000000000000003
00040016001800000000000000003
00040016001700220000000000004
00100011000000000000000000002
00100011000000000000000000002
00010004000600090022000000005
00040011001600180021000000005
1.0001.0001.0001.0001.0000.000
1.0001.0001.0000.0000.0000.000
1.0001.0001.0000.0000.0000.000
1.0001.0001.0001.0000.0000.000
1.0001.0001.0001.0001.0000.000
1.0001.0001.0000.0000.0000.000
1.0001.0001.0001.0000.0000.000
1.0001.0000.0000.0000.0000.000
1.0001.0001.0001.0000.0000.000
1.0001.0001.0001.0000.0000.000
1.0001.0001.0001.0001.0001.000
1.0001.0001.0000.0000.0000.000
1.0001.0000.0000.0000.0000.000
1.0001.0001.0000.0000.0000.000
1.0001.0001.0000.0000.0000.000
1.0001.0001.0000.0000.0000.000
1.0001.0001.0000.0000.0000.000
1.0001.0001.0001.0000.0000.000
1.0001.0000.0000.0000.0000.000
1.0001.0000.0000.0000.0000.000
1.0001.0001.0001.0001.0000.0000
1.0001.0001.0001.0001.0000.000
```

Appendix B

The Mortran Program Implementing the Graph Decomposition Method of Chapter 7

(Mortran is a structured version of Fortran)

```
%15
%Q1
"
"
"            A GRAPH DECOMPOSITION TECHNIQUE
"                    BASED ON
"            A HIGH-DENSITY CLUSTERING MODEL
"                    ON GRAPHS
"
"
"
"  INITIALIZATION MACROS
"
%'$NODES'='400'     "THE NUMBER OF NODES IN THE GRAPH
%'$MAXLINKS' = '4'  "THE MAXIMUM NUMBER OF LINKS TO A NODE
%'$COLLINKS' = '5'  "$MAXLINKS + 1
%'$INNODE' = '1'    "STARTING NODE FOR THE MINIMUM SPANNING TREE
%'$LEN' = '2'       "THE LENGTH OF EACH BRANCH OF THE CLUSTER TREE
                    "LIMIT IS $LEN * <NUMBER OF LEVELS> .LE. 100
"
LOGICAL FOUND;      "FLAG VARIABLE
"
"  INITIALIZE ARRAYS
"
DIMENSION LINKS<$NODES,$COLLINKS>; "LINKS<I,J> ARE POINTERS TO
                                   "NODES LINKED TO NODE I,
                                   "J=1,2,...$MAXLINKS
                                   "LAST COLUMN IS NUMBER OF LINKS
                                   "TO NODE I
DIMENSION WT<$NODES,$MAXLINKS>;    "WEIGHTS OF THE LINKS BETWEEN
                                   "NODE I AND NODE LINKS<I,J>
DIMENSION DC<$NODES,$MAXLINKS>;    "DENSITY CONTOURS BETWEEN NODE I
                                   "AND NODE LINKS<I,J>
DIMENSION IPATH<$NODES>;           "PATH FOR MINIMUM SPANNING TREE
DIMENSION IPATH2<$NODES>;          "NEXT NODE FOLLOWING IPATH<J>
DIMENSION PATHDC<$NODES>;          "DENSITY CONTOUR AT PATH<I>
                                   "IN MINIMUM SPANNING TREE
DIMENSION ISTACK<$NODES>;          "STACK FOR BUILDING THE
                                   " CLUSTER TREE
CHARACTER*1 ITREE<$NODES,80>;      "MATRIX TO PRINT THE
                                   " CLUSTER TREE
CHARACTER*1 ITREE1<$NODES,20>;     "AUXILLARY MATRIX TO
                                   " CONTINUE PRINTING THE
                                   " CLUSTER TREE
CHARACTER*1 ISYMS<3>;              "SYMBOLS TO DRAW THE
                                   " CLUSTER TREE
COMMON/TREE/ITREE,ITREE1,ISYMS;
"
"  INITIALIZE VARIABLES
"
%F
      ISYMS<1> = " "
      ISYMS<2> = "_"
```

```
      ISYMS<3> = "1"
%M
N = $NODES;     "THE NUMBER OF NODES IN THE GRAPH
ICOUNT = $COLLINKS;    "COLUMN FOR NUMBER OF LINKS TO NODE I
N1 = $INNODE;          "STARTING NODE FOR MINIMUM SPANNING TREE <MST>
"
"   INITIALIZE DENSITY CONTOUR MATRIX
"
<I=1,N; <J=1,$MAXLINKS;  DC<I,J> = 0.0;> >
"
"   READ IN THE LINK POINTER MATRIX
"
INPUT < < LINKS<I,J>, J=1,$COLLINKS >, I=1,N >;  <$COLLINKSI4>;
"
"   READ IN THE WEIGHTED LINKS MATRIX
"
INPUT < < WT<I,J>, J=1,$MAXLINKS >, I=1,N>;  <$MAXLINKSF5.3>;
"
"
"   COMPUTE THE DENSITY CONTOURS FOR THE LINKED NODES
"
<I=1,N;                " PROCESS ALL NODES
  FOR J=1 TO LINKS<I,ICOUNT>  " COMPUTE DENSITY CONTOUR FOR ALL
                              " NODES LINKED TO NODE I
    <N2 = LINKS<I,J>;    " N2 IS LINKED TO NODE I
     ISECT = 0;          " NUMBER OF NODES LINKED TO BOTH I AND N2
     XSECT = 0.0;        " SUM OF WEIGHTED LINKS OF NODES LINKED TO
                         " NODES I AND N2
     L1 = 1;             " POINTER TO FIRST NODE TO CHECK WHICH IS
                         " CONNECTED TO N2
     :k:
     FOR K = 1 TO LINKS<I,ICOUNT>  " CHECK IF OTHER NODES CONNECTED
                                   " TO I ARE ALSO CONNECTED TO N2
       <IF LINKS<I,K> .EQ. N2
          <NEXT :K: ;>             " ALREADY KNOWN TO BE CONNECTED
                                   " SO SKIP IT
        :L:
        FOR L = L1 TO LINKS<N2,ICOUNT> " CHECK REMAINING NODES LINKED
                                       " TO N2 FOR A MATCH
          <IF LINKS<N2,L> .GT. LINKS<I,K>
             <NEXT :K: ;>  " SINCE REST OF NODES LINKED TO N2 ARE
                           " NUMBERED EVEN HIGHER
           ELSEIF LINKS<N2,L> .EQ. LINKS<I,K>  " NODE LINKED TO BOTH
                                               " I AND N2
             <ISECT = ISECT + 1;
              XSECT = XSECT + WT<I,K§ + WT<N2,L>;
              L1 = L1 + 1;   " CHECK ONLY NODES AFTER THE MATCH
              IF L1 .GT. LINKS<N2,ICOUNT> " NO MORE NODES TO CHECK
                < EXIT :K: ;>
              EXIT :L: ;        " SINCE MATCHED THIS ONE
             > " END ELSEIF
          > " NEXT :L:
        IF LINKS<N2, LINKS<N2,ICOUNT> > .LT. LINKS<I,K>
          < EXIT :K: ; >  " NO MORE MATCHES, HIGHEST NODE LINKED TO N2
                          " LESS THAN PRESENT NODE LINKED TO I
       > " NEXT K
     UNION = LINKS<I,ICOUNT> + LINKS<N2,ICOUNT> - ISECT;
```

```
      "UNION = NUMBER OF NODES CONNECTED TO NODE I + NUMBER CONNECTED
      "TO NODE N2 - NUMBER CONNECTED TO BOTH SINCE DOUBLE COUNTED
      XSECT = .5 * XSECT + 2. * WT<I,J>,
      DC<I,J> = XSECT/UNION; "DENSITY CONTOUR FOR NODES I AND N2
     > "NEXT J
> "NEXT I
"
"   FIND THE MINIMUM SPANNING TREE <MST>
"
FOR I = 1 to N-1
  <IPATH<I> = N1;        " ITH NODE OF MST
   DMAX = 0.0;           " INITIALIZE MAX DENSITY CONTOUR
   K = 1;                " INITIALIZE STARTING AT THE ROOT
   WHILE K .LE. I        " CHECK FOR MAX DENSITY CONTOUR FROM CLUSTER
                         " FORMED THUS FAR TO NEXT NODE IN THIS TREE
     <N1 = IPATH<K>;
      FOR J = 1 TO LINKS<N1,ICOUNT>
        <IF DC<N1,J> .GT. DMAX        "THEN THERE IS A NEW MAX DC
           <N2 = LINKS<N1,J>;         "MAX IS FOR NODE N2
            DMAX = DC<N1,J>;          "SAVE NEW MAX
           > "END IF
        > "NEXT H
      K = K + 1;       " CONTINUE DOWN THE TREE
     > " END WHILE
   IPATH2<I> = N2;   " NEXT NODE IN THE MST
   PATHDC<I> = DMAX;  "DENSITY CONTOUR FOR THE TREE BRANCH
   K = 1;    " REINITIALIZE AND REMOVE CONNECTED NODES' DC'S FROM
             " FURTHER CONSIDERATION
   WHILE K .LE. I
     <N1 = IPATH<K>;
      FOUND = .FALSE. ;  " NO MATCH FOUND YET
      ;LI;
      FOR L = 1 TO LINKS<N2,ICOUNT>
        <IF LINKS<N2,L> .EQ. N1      " FOUND A MATCH
           <DC N2,L  = -DC<N2,L>; FOUND = .TRUE. ; EXIT :LI: ;>
        > "NEXT L1
      IF FOUND         " MATCH FOUND SO PROCESS N1
        <:L2:
         FOR L = 1 TO LINKS<N1,ICOUNT>
           <IF LINKS<N1,L> .EQ. N2     " FOUND THE NODE
              <DC<N1,L> = -DC<N1,L>; EXIT :L2: ; >
           > " NEXT L2
        > " END IF
      K = K + 1;        " CONTINUE DOWN THE NODES OF THE TREE
     > " END WHILE
   N1 = N2;           " NEXT NODE IN THE TREE
> " NEXT I
PATHDC<N> = 0.0;       " SET FOR EASE OF OUTPUT
IPATH<N> = N1;         " LAST NODE IN THE TREE
IPATH2<N> = 0;         " FOR EASE OF OUTPUT
"
"   OUTPUT THE MINIMUM SPANNING TREE
"
OUTPUT; <///'              THE MINIMUM SPANNING TREE'//>;
OUTPUT < IPATH<I>,IPATH2<I>,PATHDC<I>, I = 1,N>;
<'  NODE',I4,' CONNECTED TO NODE',I4,' WITH DENSITY CONTOUR =',F6.4>;
"
```

```
"
"   BUILD THE CLUSTER TREE
"
"
"
"
"   INITIALIZE THE CLUSTER TREE MATRIX TO BLANKS
"
<I=1,N;  <J=1,80; ITREE<I,J> = ISYMS<1>;> >
<I=1,N; <J=1,20; ITREE1<I,J> = ISYMS<1>;> >
"
"   INITIALIZE VARIABLES
"
IPTR = 0;      "STACK POINTER
LEVT = 1;      "LEVEL OF THE UPPER NODE OF THE TREE TO BE CONNECTED
LEVB = 1;      "LEVEL OF THE LOWER NODE OF THE TREE TO BE CONNECTED
K = $LEN;      "THE LENGTH OF EACH BRANCH OF THE CLUSTER TREE
ITOP =1;       "START AT THE TOP OF IPATH
LPCNTR = 0;    "LOOP COUNTER FOR DEBUGGING AND STATS
"
:LP:           "MAIN LOOP FOR BUILDING THE CLUSTER TREE
LOOP
  <I = ITOP;
   WHILE ABS<PATHDC<I> - PATHDC<I+1>> .LT. 1.E-9  <I=I+1;>
   "WHILE LOOP CHECKS FOR SEQUENCES OF EQUAL DENSITY CONTOUR
   "VALUES TO BE CONNECTED
   IF PATHDC<I> .GT. PATHDC<I+1>  "CONNECT ALL NODES TO THIS POINT
     <IBOT = I+1;  MAXLEV = MAX0<LEVT,LEVB>;
      CALL BUILDT<ITOP,IBOT,MAXLEV,LEVT,LEVB,K>;
             "BUILDT CONNECTS THE TWO NODES
      ITOP = IBOT; "MAKE BOTTOM NODE CONNECTED THE CURRECT TOP
      LEVT = MAXLEV + 1; "ADJUST LEVEL OF NEW TOP NODE
     > "END IF
   WHILE IPTR .NE. 0     " WHILE STACK ISN'T EMPTY, CHECK BACK
     <IBOT = ITOP; ITOP = ISTACK<IPTR-1>; "MAKE TOP OF STACK THE
                                          "NEW TOP NODE AND
      LEVB = LEVT; LEVT = ISTACK<IPTR>;   "THE OLD TOP NODE THE
                                          "NEW BOTTOM NODE
      IF PATHDC<ITOP> .GT. PATHDC<IBOT>   "CONNECT THE TWO
        <MAXLEV = MAX0<LEVT,LEVB>;  "HIGHEST LEVEL BETWEEN THE NODES
         CALL BUILDT<ITOP,IBOT,MAXLEV,LEVT,LEVB,K>
         ITOP = IBOT; LEVT = MAXLEV + 1;  "NEW TOP NODE
         IPTR = IPTR - 2;   "POP THE STACK
        > "END IF
      ELSE <ITOP = IBOT; LEVT = LEVB; EXIT "WHILE";> "RESET PREV TOP
     > "END WHILE
   IF PATHDC<ITOP> .EQ. 0.0    "WE'RE DONE BUILDING THE TREE
     <EXIT :LP: ;>
   ELSEIF PATHDC<ITOP> .LT. PATHDC<I+1>   "ADD ITOP TO THE STACK
     <IPTR = IPTR+1; ISTACK<IPTR> = ITOP;  "PUSH ITOP ONTO STACK
      IPRT = IPTR+1; ISTACK<IPTR> = LEVT;  "SAVE ITS LEVEL ALSO
      ITOP = ITOP + 1; LEVT = 1;  "CONTINUE DOWN IPATH
     > "END ELSEIF
   LEVB = 1;   "NEW BOTTOM NODE NOT YET CONSIDERED
   LPCNTR = LPCNTR + 1;   "INCREMENT THE LOOP COUNTER FOR DEBUGGING
   IF LPCNTR .GT. 10000  "MOST BE IN AN INFINITE LOOP
     <OUTPUT; <' INFINITE LOOP -- CHECK THE DC VALUES'///>;EXIT;>
```

```
  > "END LOOP :LP:
"
"
"   PRINT THE CLUSTER TREE
"
OUTPUT; <////'           THE CLUSTER TREE '//>;
"
OUTPUT <IPATH<I>,PATHDC<I>,<ITREE<I,J>,J=1,80>,<ITREE1<I,J>,J=1,20>,
        I=1,N>;
<' NODE',14,  DC',F6,4,1X,100A1>;
"
"
STOP; END;
"
"
"
"*************************************************************
"*************************************************************
"
"
"   SUBROUTINE FOR CONNECTING NODES TO BUILD THE CLUSTER TREE
"
"
SUBROUTINE BUILDT<ITOP,IBOT,MAXLEV,LEVT,LEVB,K>;
"
"   DECLARE THE ARRAYS
"
CHARACTER*1 ITREE<$NODES,80>;   " THE CLUSTER TREE
CHARACTER*1 ITREE1<$NODES,20>;  " CONTINUATION OF CLUSTER TREE
CHARACTER*1 ISYMS<3>;          THE TREE SYMBOLS
COMMON/TREE/ITREE,ITREE1,ISYMS;
"
LEND = K * MAXLEV;     " THE END POINT OF THE CONNECTION
IF LEND .GT. 100   " CLUSTER TREE EXCEEDS MATRIX LIMITS
  <OUTPUT; <//'EXCEEDED 100 COLUMNS IN THE CLUSTER TREE'>;
  Note: this is an HP3000 limitation
   OUTPUT MAXLEV; </  MAXLEV = ',14>;
   RETURN;
  > " END IF
IF LEND .LE. 80   "ONLY ITREE INVOLVED
  <FOR J = K * <LEVT-1> + 1  TO  LEND-1   " MAKE BRANCH TO TOP NODE
     <ITREE<ITOP,J> = ISYMS<2>;>
   FOR J = K * <LEVG-1> + 1  TO  LEND-1   " MAKE BRANCH TO BOTTOM NODE
     <ITREE<IBOT,J> = ISYMS<2>;>
   FOR J = ITOP+1 TO IBOT
     <ITREE<J,LEND> = ISYMS<3>;>   " CONNECT THE BRANCHES
  > " END IF
ELSE   " TREE EXCEEDS LIMITS OF ITREE MATRIX
  " MAKE BRANCH TO TOP NODE
  <IF LEVT .LT. 41   "MODIFY ITREE AND ITREE1
     <FOR J = K*<LEVT-1> + 1 TO 80
        <ITREE<ITOP,J> = ISYMS<2>;>
      FOR J = 81 TO LEND-1
        <ITREE1<ITOP,J-80> = ISYMS<2>;>
     >
   ELSE   "PASSED ITREE, FILL IN ITREE1 ONLY
     <FOR J = K*<LEVT-1> + 1 TO LEND-1
        <ITREE1<ITOP,J-80> = ISYMS<2>;>
```

```
      >
    " MAKE BRANCH TO BOTTOM NODE
    IF LEVB .LT. 41   " MODIFY ITREE AND ITREE1
      <FOR J = K*<LEVB-1> + 1 TO 80
         <ITREE<IBOT,J> = ISYMS<2>;>
       FOR J = 81 TO LEND-1
         <ITREE1<IBOT,J-80> = ISYMS<2>;>
      >
    ELSE   " PASSED ITREE, FILL IN ITREE1 ONLY
      <FOR J = K*<LEVB-1> + 1 TO LEND-1
         <ITREE1<IBOT,J-80> = ISYMS<2>;
      >
   " CONNECT THE TWO BRANCHES
    FOR J = ITOP+1 TO IBOT
      <ITREE1<J,LEND-80> = ISYMS<3>;>
  >   " END ELSE
RETURN; END;
%%
```

Acknowledgements

I would like to acknowledge my appreciation to Ms. Phyllis Ruby for her meticulous typing of this thesis.

I am deeply indebted to Professor M. Anthony Wong for his suggestions, criticisms, and invaluable guidance throughout my research.

In addition, a special thanks to my wife, Cindy, and my children, Sean and Shannon, for their patience, encouragement and understanding. Finally, my wife is due an extra thanks for the many hours she spend copying and proofreading and constucting the many figures.

Bibliography

Anderberg, M. R. (1973). Cluster Analysis for Applications. New York: Academic Press.

Andreu, R. (1978). A Systematic Approach to the Design and Structure of the Complex Software Systems. Ph.D. Dissertation, Sloan School of Management, Massachusetts Institute of Technology.

Blashfield, R. K., and Aldenderfer, M. S. (1978). "The Literature on Cluster Analysis". Multivariate Behavioral Research, 13, 271-295.

Carmichael, J. W., George, J. A., and Julius, R. S. (1968). "Finding natural clusters". Systematic Zoology, 17, 144-150.

Cormack, R. M. (1971). "A Review of Classification". Journal of the Royal Statistical Society, Series A, 134, 321-367.

Devroye, L. P. and Wagner, T. J. (1977). "The strong uniform consistency of nearest neighbor density estimates". Annals of Statistics, 5, 536-540.

Engelman, L., and Hartigan, J. A. (1969). "Percentage Points of a Test for Clusters". Journal of the American Statistical Association, 64, 1647-1648.

Everitt, B. S. (1974). Cluster Analysis, Halsted Press, New York: John Wiley.

Friedman, H. P., and Rubin, J. (1967). "On some invariant criteria for grouping data". Journal of the American Statistical Association, 62, 1159-1178.

Friedman, J. H., Bentley, J. L., and Finkel, R. A. (1975). "An algorithm for finding best matches in logarithmic time." SLAC-PUB-1549, Feb., 1975.

Gower, J. C. and Ross, G. J. S. (1969). "Minimum spanning trees and single linkage cluster analysis". Applied Statistics, 18, 54-64.

Hartigan, J. A. (1975). Clustering Algorithms. New York: John W-ley & Sons.

Hartigan, J. A. (1977a). "Distributional problems in clustering", in Classification and Clustering. ed. J. Van Ryzin, New York: .Academic Press.

Hartigan, J. A. (1977b). "Clusters as modes", in First International Symposium on Data Analysis and Informatics, Vol. 2, IRIA, Versailles.

Hartigan, J. A. (1978). "Asymptotic distributions for clustering criteria". Annals of Statistics, 6, 117-131.

Hartigan, J. A. (1979). "Consistency of single linkage for high-density clusters". Unpublished manuscript, Department of Statistics, Yale University.

Hartigan, J. A. and Wong, M. A. (1979). "Algorithm AS136: A K-means Clustering Algorithm". Applied Statistics, 28, 100-108.

Hartigan, J. A. (1981). "Consistency of single linkage for high-density clusters". Journal of the American Statistical Association, 76, 388-394,

Kauth, R. J. and Thomas, G. S. (1976). "The Tasselled Cap. Proceedings of the Symposium on Machine Processing of Remotely Sensed Data". LARS, Purdue University.

Lee, K. L. (1979). "Multivariate Tests for Clusters". Journal of the American Statistical Association, 74, 708-714.

Misra, P. N. and Wheeler, S. G. (1978). "Crop Classification with Landsat Multispectral Scanner Data". Pattern Recognition, 10, 1-13.

Misra, P. N. and Wheeler, S. G. (1980). "Crop Classification with Landsat Multispectral Scanner Data II". Pattern Recognition, 12, 219-228.

"Proceedings of the Symposium on Large Area Crop Inventory Experiment (LACIE)". (1978). National Aeronautics and Space Administration. Lyndon B. Johnson Space Center, Houston, Texas.

Sneath, P. H. A. (1957). "The application of computers to taxonomy". Journal of General Microbiology, 17, 201-226.

Sneath, P. H. A. and Sokal, R. R. (1973). Numerical Taxonomy. San Francisco: W. H. Freeman.

Sorenson, T. (1948). "A method of estimating groups of equal amplitude in plant sociology based on similarity of species content". K. Danske Vidensk. Selsk. Skr. (Biol.), 5, 1-34.

Spath, H. (1980). Cluster Analysis Algorithms. Halsted Press, New York: John Wiley.

Wishart, D. (1969). "Mode Analysis" in Numerical Taxonomy. edited by A. J. Cole, New York: Academic Press.

Wong, M. A. (1982). "A hybrid clustering method for identifying high-densith clusters". Journal of the American Statistical Association, 77, 841-847.

Wong, M. A. and Lattin, J. M. (1982). "A graph decomposition procedure based on the single linkage clustering method". Classification Society Bulletin. (To appear).

Wong, M. A. and Schaack, C. (1982). "Using the Kth Nearest Neighbor Clustering Procedure to Determine the Number of Subpopulations". Proceedings of the American Statistical Association, Statistical Computation Section.

Wong, M. A. and Lane, T. (1983). "A Kth Nearest Neighbor Clustering Procedure". Journal of the Royal Statistical Society, Series B, (to appear).


END
FILMED
4-84
DTIC